

**UNIVERSIDAD MAYOR DE SAN ANDRES  
FACULTAD DE CIENCIAS PURAS Y NATURALES  
CARRERA INFORMATICA**



**PROYECTO DE GRADO**

**GESTION DEL PROCESO PRODUCTIVO  
DE LA FABRICA “INTIWARA”**

**PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA**

Postulante: Guillermo Nicanor Tola Antiñapa

Tutor: Lic. Efraín Silva Sánchez

Revisor: Lic. Franz Cuevas Quiroz

LA PAZ – BOLIVIA  
2011

## DEDICATORIA

Al hombre que me enseñó los valores más importantes, el trabajo y la honestidad,  
mi padre Alberto (†).

A mi madre Andrea, por su apoyo incondicional y su infinito cariño.

A mi esposa Elsa, la mujer que amo, por su comprensión, apoyo y amor.

A mis hijos, Génesis, Daniel, Pablo y Samuel, por su cariño y porque ellos son mi  
razón y mi inspiración en la vida.

A mis hermanos, Edgar, Betza (†), Ramiro, Adolfo, Antonia y José Luis por su  
apoyo y comprensión.

## AGRADECIMIENTOS

Doy gracias a mi Dios, por sus bendiciones y su cuidado, por sus grandes milagros en mi vida, porque gracias a Él logré concluir mi carrera.

Gracias, al Lic. Efraín Silva y Lic. Franz Cuevas por su apoyo y comprensión, por las correcciones y sugerencias.

Gracias, al Sr. Nino Molinari y a la Sra. Elena Cirbian, propietarios de la fábrica “Intiwara”, por su apoyo y colaboración.

## INDICE GENERAL

	Pag.
Capítulo I Presentación	
1. Presentación .....	1
1.1. Introducción .....	2
1.2. Antecedentes .....	3
1.3. Formulación del Problema.....	5
1.4. Objetivos .....	6
1.4.1. Objetivo General.....	7
1.4.2. Objetivos Específicos .....	7
1.5. Justificación .....	7
1.6. Alcances y Límites.....	8
1.7. Metodología .....	9
Capítulo II Marco Teórico	
2. Marco Teórico .....	10
2.1. Software .....	11
2.2. Características del Software.....	12
2.3. Ingeniería de Software.....	12
2.4. Ciclo de Vida del Software .....	13
2.4.1. Ciclo de vida Clásico .....	13
2.4.2. Modelo Espiral .....	15
2.5. Ingeniería de Requerimientos .....	18
2.5.1. Concepción .....	18
2.5.2. Indagación .....	18
2.5.3. Elaboración .....	19
2.5.4. Negociación .....	19
2.5.5. Especificación .....	19
2.5.6. Validación .....	20
2.5.7. Administración de Requerimientos.....	20
2.6. Proceso Unificado de Desarrollo .....	20
2.6.1. Dirigido por Casos de Uso.....	21
2.6.2. Centrado en la Arquitectura.....	21
2.6.3. Iterativo e Incremental .....	22

2.6.4. La Vida del Proceso Unificado .....	22
2.6.5. Flujos de Trabajo .....	24
2.7. UML.....	24
2.7.1. Casos de Uso .....	25
2.7.2. Diagramas de Clases .....	26
2.7.3. Diagrama de Transición de Estados .....	27
2.7.4. Diagramas de Secuencia .....	28
2.8. Base de Datos .....	29
2.8.1. Base de Datos Relacional .....	29
2.8.2. Gestor de Base de Datos .....	29
2.8.3. Gestor de Base de Datos MySql.....	30
2.9. PHP .....	32
2.9.1. Características de PHP .....	32
2.10. Arquitectura Cliente/Servidor.....	33
2.10.1. Cliente .....	34
2.10.2. Servidor.....	35
2.10.3. Características de la Arquitectura Cliente/Servidor .....	35
2.10.4. Ventajas de la Arquitectura Cliente /Servidor .....	36
2.10.5. Desventajas de la Arquitectura Cliente /Servidor .....	37
2.11. Planificación.....	38
2.12. Control.....	39
2.13. Relación Entre Planificación y Control .....	40
2.14. Teoría de Inventarios.....	41
2.14.1. Modelo del Lote Económico de Pedido (EOQ) .....	41
2.15. Algoritmos Voraces.....	43
2.16. Calidad del Software.....	44
2.16.1. Factores de Calidad ISO 9126.....	45
2.17. Seguridad.....	46
Capítulo III Marco Aplicativo	
3. Marco Aplicativo.....	48
3.1. Antecedentes .....	49
3.2. Análisis del Sistema Actual .....	51
3.3. Diseño del Nuevo Sistema .....	56
3.3.1. Especificación de Requerimientos .....	56

3.3.2. Casos de Uso .....	58
3.3.3. Arquitectura del Sistema .....	62
3.3.4. Diagramas de Clases .....	63
3.3.5. Diagrama de Transición de Estados .....	64
3.3.6. Diagrama de Secuencia .....	67
3.4. Diseño de Interfaz.....	69
3.5. Requerimientos para la Implementación.....	76
Capítulo IV Calidad y Seguridad	
4. Calidad y Seguridad .....	77
4.1. Calidad del Sistema.....	78
4.1.1. Funcionalidad.....	78
4.1.2. Usabilidad .....	79
4.1.3. Eficiencia.....	79
4.1.4. Facilidad de Recibir Mantenimiento .....	80
4.1.5. Portabilidad .....	81
4.2. Seguridad del Sistema.....	81
4.2.1. Seguridad de Acceso.....	81
4.2.2. Autenticación de Usuarios.....	82
Capítulo V Conclusiones y Recomendaciones	
5. Conclusiones y Recomendaciones .....	83
5.1. Conclusiones .....	84
5.2. Recomendaciones .....	85
Bibliografía .....	86
Anexos .....	87

## LISTA DE FIGURAS

Figura 2.1. El Modelo Lineal Secuencial.....	15
Figura 2.2 El Modelo Espiral .....	16
Figura 2.3. Proceso de Desarrollo de Software.....	20
Figura 2.4. Un Ciclo con sus Fases e Iteraciones.....	22
Figura 2.5. Flujos de Trabajo y Fases .....	24
Figura 2.6. Diagrama de Casos de Uso.....	26
Figura 2.7. Diagrama de Clases .....	27
Figura 2.8. Diagrama de Transición de Estados .....	28
Figura 2.9. Diagrama de Secuencia.....	29
Figura 2.10. Modelo Cliente/Servidor .....	34
Figura 2.11. Relación entre la Planificación y el Control.....	41
Figura 2.12. Modelo del Lote Económico de Pedido.....	43
Figura 3.1. Organigrama de la Fabrica “Intiwara”.....	50
Figura 3.2. Caso de Uso – Gestión de Archivos.....	59
Figura 3.3. Caso de Uso – Planificación de la Producción .....	59
Figura 3.4. Caso de Uso – Control de la Producción .....	60
Figura 3.5. Caso de Uso – Control de Almacenes .....	61
Figura 3.6. Arquitectura del Sistema.....	62
Figura 3.7. Organización del Sistema.....	62
Figura 3.8. Diagrama de Clases .....	63
Figura 3.9. Diagrama Transición de Estados – Registro de Pedidos .....	64
Figura 3.10. Diagrama Transición de Estados – Planificación.....	64
Figura 3.11. Diagrama Transición de Estados – Asignación de Trabajo.....	65
Figura 3.12. Diagrama Transición de Estados – Recepción de Trabajo .....	65
Figura 3.13. Diagrama Transición de Estados – Entrega de Materia Prima.....	66
Figura 3.14. Diagrama Transición de Estados – Pedido de Materia Prima.....	66
Figura 3.15. Diagrama de Secuencia – Planificación.....	67
Figura 3.16. Diagrama de Secuencia – Asignación de Producción .....	68
Figura 3.17. Diagrama de Secuencia – Recepción de Producción.....	68
Figura 3.18. Diagrama de Secuencia – Entrega de Materia Prima.....	69
Figura 3.19. Diseño de Interfaz – Pantalla de Acceso .....	69
Figura 3.20. Diseño de Interfaz – Menú Principal.....	70
Figura 3.21. Diseño de Interfaz – Registro de Productos.....	72

Figura 3.22. Diseño de Interfaz – Registro de Personal.....	73
Figura 3.23. Diseño de Interfaz – Registro de Usuarios.....	73
Figura 3.24. Diseño de Interfaz – Tabla de Medidas.....	74
Figura 3.25. Diseño de Interfaz – Planificación de la Producción.....	74
Figura 3.26. Diseño de Interfaz – Asignación de Tejido.....	75
Figura 3.27. Diseño de Interfaz – Recepción de Tejido .....	75
Figura 3.28. Diseño de Interfaz – Reporte de Producción .....	76



## LISTA DE TABLAS

Tabla 3.1. Recepción de Pedidos .....	51
Tabla 3.2. Proceso Productivo – Tejido .....	52
Tabla 3.3. Proceso Productivo – Costura .....	53
Tabla 3.4. Proceso Productivo – Planchado.....	54
Tabla 3.5. Proceso Productivo – Control de Calidad y Acabado .....	55
Tabla 3.6. Especificación de Requerimientos – Gestión de Archivos.....	57
Tabla 3.7. Especificación de Requerimientos – Planificación de la Producción.....	57
Tabla 3.8. Especificación de Requerimientos – Control de la Producción .....	57
Tabla 3.9. Especificación de Requerimientos – Control de Almacenes .....	58
Tabla 3.10. Especificación de Requerimientos – Reportes.....	58
Tabla 4.1. Funcionalidad del Sistema.....	79
Tabla 4.2. Usabilidad del Sistema.....	79
Tabla 4.3. Uso de Recursos.....	80
Tabla 4.4. Facilidad de Mantenimiento .....	80

## RESUMEN

El presente proyecto contiene el desarrollo del sistema “Gestión del Proceso Productivo de la Fábrica Intiwara”, que comprende los procesos administrativos y transacciones realizadas en el departamento de producción de la indicada fábrica, poniendo énfasis en la planificación y control de la producción, considerando también la gestión de almacenes como un aspecto importante dentro el proceso productivo.

Al inicio se presenta las generalidades del proyecto de grado, se describe los antecedentes de la fábrica y algunos proyectos realizados anteriormente, se identifica los problemas de la empresa, los objetivos propuestos en respuesta a los problemas identificados anteriormente, la justificación del proyecto, los alcances, límites y la metodología de desarrollo.

Para la especificación de los requerimientos y el diseño del sistema se realizó un estudio de las diferentes actividades realizadas en las diferentes secciones del departamento de producción.

El presente sistema fue desarrollado siguiendo los lineamientos propuestos por la Ingeniería de Software y factores de calidad necesarios para el buen desarrollo del proyecto, usando la metodología planteada en el modelo espiral evolutivo. Para el desarrollo del software se utilizó el Proceso de Desarrollo Unificado RUP, junto con el Lenguaje de Modelado UML, teniendo en cuenta los requerimientos establecidos con anterioridad.

Este proyecto plantea la implementación de un software diseñado bajo la arquitectura cliente – servidor con una base de datos relacional, utilizando para el desarrollo de la interfaz de usuario HTML, CSS y Flash, como gestor de base de datos se usó MySql y lenguaje de programación PHP. Los conceptos y características de las técnicas o herramientas usadas en el presente proyecto son descritos en el segundo capítulo.

Finalmente se presenta las conclusiones a las que se llegó después de haber concluido el proyecto y algunas recomendaciones para la ampliación del sistema.

# **1. PRESENTACIÓN**



## 1. PRESENTACIÓN

### 1.1. Introducción

En la administración de entidades productivas una adecuada gestión de los procesos realizados es una de las tareas más importantes que se realizan, en tal sentido la planificación y el control de las actividades realizadas son tareas básicas en la gestión empresarial.

Una planificación adecuada es el elemento clave para alcanzar los objetivos propuestos y si esta planificación viene acompañada de un control adecuado y oportuno en las actividades planificadas, el logro del objetivo será alcanzada fácilmente, en cambio si se propone alcanzar un objetivo sin una planificación y menos sin contar con los controles necesarios los resultados muy probablemente no serán los esperados.

En instituciones en las que se deben cumplir con ciertas restricciones que tengan que ver, con requisitos a cumplir, tiempos que satisfacer o logros a alcanzar, una planificación precisa es un requisito imprescindible que se debe cumplir para satisfacer dichas restricciones, esta planificación debe ir acompañada de los controles necesarios para evaluar si se están cumpliendo adecuadamente con las restricciones mencionadas o si por el contrario no se cumplen.

La facilidad con la que ahora se cuenta para acceder a recursos que brinda la tecnología informática para facilitar la planificación y el control dentro de las instituciones y así poder lograr mayores beneficios, hace que se plantee el siguiente proyecto de grado para poder aprovechar dichos recursos en beneficio de la fabrica “Intiwara” y que repercute en las personas que son parte de dicha institución.

La implementación del presente proyecto busca asistir al personal administrativo en la gestión del proceso productivo, poniendo énfasis en la planificación y control de dicho proceso.

## **1.2. Antecedentes**

La fabrica “Intiwara” se encuentra ubicada en la zona de Sopocachi de la ciudad de La Paz y se dedica a la confección y comercialización de chompas de alpaca, realizando también otros tejidos (guantes, chalinas, abrigos, etc.) en menor cantidad, la comercialización de los productos confeccionados se realiza principalmente en el exterior del país, teniendo también puntos de venta en nuestra ciudad.

En la actualidad la fábrica cuenta con personal fijo y personal eventual que es contratado de acuerdo a la demanda de productos. También se subcontrata pequeños talleres a los cuales se proporciona materia prima para la elaboración de los diferentes productos demandados.

La planificación y el control de la producción es realizada de acuerdo a la experiencia obtenida por los dueños que a su vez se desempeñan como administradores, el registro y control de las actividades relacionadas a la producción se realiza de forma manual lo que ocasiona que no se pueda contar con información oportuna y actualizada del avance de los diferentes pedidos realizados por los clientes y que obtener dicha información representa una pérdida de tiempo y de recursos que se acentúa especialmente en época donde las ventas y pedidos alcanzan un mayor nivel.

Por la información recolectada cabe señalar que la carrera de informática cuenta en la biblioteca con documentos de trabajos relacionados con el presente tales como:

- Sistema de Apoyo a la Producción para Orma Ltda. Realizado por Raquel Carola Balderrama Ibañez – 2005.

Este sistema plantea la implementación de un sistema automatizado diseñado bajo la arquitectura cliente – servidor, utilizando como lenguaje de programación el Microsoft Visual Basic y como gestor de base de datos el SQL – Server 2000.

Además este proyecto utiliza para su desarrollo técnicas de la Ingeniería de Software, desde el uso de herramientas de modelado como ser el modelo de desarrollo incremental evolutivo.

Este proyecto se limita al tratamiento y control de los procesos de producción e inventarios y no así automatizar la producción misma.

- Sistema de Información para la Producción Planificada Delizia La Paz. Realizado por María Esther Apaza Navia – 2006.

Este proyecto de grado tiene como propósito desarrollar un sistema para planificar la producción, que permita facilitar la planificación y administración de la información generada por la compañía de alimentos DELIZIA La Paz.

Para el análisis y diseño del nuevo sistema se utilizó la metodología orientada a objetos RUP (Proceso Unificado de Rational), con el lenguaje de modelado de sistemas de software UML (Lenguaje de Modelado Unificado), para la gestión del almacenes se aplicó el método de decisión de gestión de inventarios CEP (Cantidad Económica de Pedido) y los métodos de valuación de inventarios FIFO y LIFO. El sistema funciona en la red interna de la empresa, brindando a los usuarios autorizados, reportes e informes de las demandas registradas de materia prima o insumos, espacios físicos necesarios para almacenar la materia prima, productos en cámara, productos en almacenes y determinación de los costos de producción para la toma de decisiones de las áreas respectivas.

- Sistema Integrado de Administración y Gestión Creatronica SRL. Realizado por Maribel Felizia Machicado Suculento – 2007.

Este proyecto tiene el propósito de realizar el seguimiento y control de los proyectos y equipos en reparación además de generar reportes diarios por puntos de control.

Para el análisis y diseño del sistema se usa la metodología orientada a objetos RUP junto con el lenguaje de modelado UML y factores de calidad necesarios para el buen desarrollo del proyecto.

Para el desarrollo está basado en Java Script, Ajax, PHP y HTML para la generación de la interfaz y como gestor de base de datos MySql.

Existen también sistemas que se ofertan a través de Internet como por ejemplo:

Sistema de Planificación y Control de Producción SISPRO.

Proyecto ofrecido para la venta, que propone la planificación completa de la producción, realizando un control de todas las actividades realizadas en el departamento de producción.

### **1.3. Formulación del Problema**

Luego de realizar un estudio de las actividades que involucran el proceso productivo dentro de la fábrica "Intiwara" se pudo determinar diferentes problemas, a continuación se menciona los principales:

- Existe un inadecuado registro de las transacciones que se realizan en el departamento de producción, debido a que esta tarea se lo realiza de forma manual y de manera informal, causando que la recolección de la información sea un trabajo dificultoso y moroso, y que por este motivo la toma de decisiones sea bajo un alto nivel de incertidumbre y/o que estas decisiones se vean demoradas
- La asignación de trabajo se lo realiza de acuerdo a la experiencia adquirida por los administradores, no teniendo para esta actividad un método o política que pueda garantizar los plazos establecidos, causando muchas veces el incremento de los costos de producción, pues, se debe aumentar las horas de trabajo (horas extras que se deben pagar el doble), o la contratación de personal extra, para satisfacer los tiempos.
- Otra inconveniente que se pudo observar, es que no se cuenta con normas de control que determine el avance o estado en que se encuentra un determinado pedido y que para obtener esta información se debe revisar los registros

personales o ir a preguntar a cada empleado el estado en que se encuentra el trabajo asignado, esta actividad representa pérdida de tiempo para los administradores. En algunas ocasiones se encuentra mucho atraso en el avance de un pedido y que no queda otra que posponer el envío de los productos requeridos por un cliente y que esto ocasiona pérdida de credibilidad o lo que es peor el pago de multas o sanciones.

- Debido al registro manual del movimiento de materia prima dentro de la institución, no se puede determinar oportunamente la situación en que se encuentra la materia prima, es decir, la cantidad disponible, la cantidad que se encuentra en proceso de producción, o la cantidad que se requiere para satisfacer los pedidos de los clientes, provocando una mala asignación de este recurso, causando muchas veces el retraso en el avance o entrega de los diferentes pedidos.

Por lo mencionado anteriormente y por el crecimiento de la fábrica y la necesidad de realizar un mejor control de las transacciones inherentes al proceso productivo hace que surja la necesidad de contar con métodos y herramientas que faciliten y coadyuven en la gestión del proceso productivo, haciendo énfasis en la planificación y control de la producción dentro y fuera de la fábrica.

Por lo que:

¿La planificación y control automatizado de las actividades realizadas en el departamento de producción facilitará la gestión del proceso productivo de la fábrica "Intiwara"?

#### 1.4. Objetivos

Los objetivos propuestos para el desarrollo del presente proyecto de grado son los siguientes:



#### 1.4.1. Objetivo General

Desarrollar e implementar software para facilitar la planificación y el control de la producción en la fábrica “Intiwara”.

#### 1.4.2. Objetivos Específicos

Para alcanzar el objetivo general se plantea realizar los siguientes módulos:

- Gestión de los diferentes archivos (personal, productos, pedidos, etc.).
- Planificación de la producción.
- Gestión de almacenes.
- Asignación y recepción de producción.
- Reportes de las diferentes etapas de producción.

Integrados los módulos anteriormente mencionados formarán el sistema planteado.

#### 1.5. Justificación

Para el desarrollo de este proyecto se requiere la formulación de modelos estadísticos que permitan alcanzar con los objetivos planteados anteriormente, por lo tanto se requiere un estudio de la teoría estadística para la elaboración formal de dichos modelos que puedan ser incluidos en un programa informático. También se considera la utilización de algoritmos voraces que se adecuan a los problemas encontrados y facilitan la resolución de los mismos.

Por otro lado con el apoyo del presente sistema se pretende tener una planificación que evite tener sanciones por incumplimiento de los pedidos realizados por los clientes, tener los productos solicitados listos para su envío con un tiempo de anticipación adecuado, que permita elegir la mejor ruta y/o los servicios de transporte más económicos para el envío de los productos a los diferentes clientes.

La implementación y explotación del Sistema Informático pretende brindar un apoyo para un adecuado control y así evitar la pérdida y por lo tanto una adecuada utilización

de los recursos (materiales, materia prima, tiempo, humanos, etc.). Al tener un adecuado control de los recursos de la institución esto repercute positivamente en la economía de la fábrica, lo que representa mayores beneficios para los empleados y por consiguiente para sus respectivas familias.

Al facilitar la planificación y el control del cumplimiento de esta, se evita los apuros que se tiene por una inadecuada planificación y un control deficiente que provoca estrés en todos los miembros de la fábrica por la presión de cumplir con los clientes.

Por lo anteriormente mencionado el presente proyecto queda plenamente justificado.

## **1.6. Alcances y Limites**

El Presente proyecto de Grado se circunscribe a la fábrica “Intiwara”, es decir, se encuentra limitado a cubrir las necesidades de control de la mencionada fábrica. Por otro lado se plantea alcanzar los siguientes aspectos:

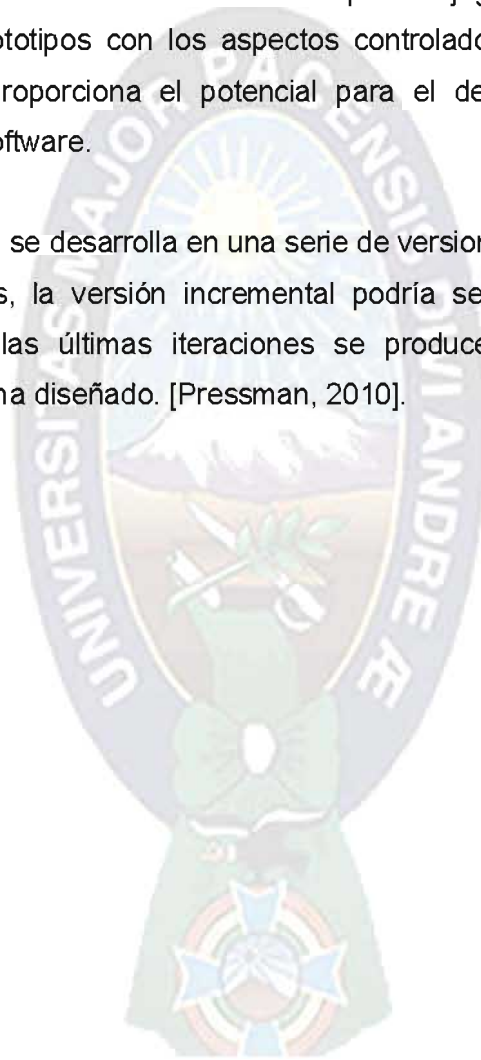
- Realizar una planificación de los pedidos recibidos generando diferentes alternativas que permitan cumplir con los plazos establecidos, estas opciones pueden ser el incremento de horas extras de trabajo, contratación de más empleados, etc. Presentando por cada alternativa una posible planificación que permita cumplir con los pedidos realizados.
- Coadyuvar en la gestión de almacenes que permita tener un cálculo aproximado de la materia prima requerida para cada pedido, la materia prima existente, la cantidad óptima de material pedido y la generación de órdenes de compra con la debida anticipación.
- Tener una gestión de alarmas tempranas cuando la planificación no se cumpla por problemas que se puedan presentar en las diferentes secciones (tejido, costura, planchado, bordado, acabado y control de calidad) involucradas en el proceso productivo o cuando la falta de algún recurso (empleados, materia prima, herramientas) pueda interferir en dicha planificación.
- Brindar información actualizada y oportuna del avance de un determinado pedido en las diferentes secciones y la disponibilidad de materia prima para una adecuada

toma de decisiones, para un uso conveniente de los recursos que permita cumplir con la planificación en los plazos establecidos.

### **1.7. Metodología**

La metodología que se plantea para el análisis diseño e implementación del presente proyecto es el modelo en espiral propuesto originalmente por BOEHM [BOE88], es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de construcción de prototipos con los aspectos controlados y sistemáticos del modelo lineal secuencial, proporciona el potencial para el desarrollo rápido de versiones incrementales del software.

En el modelo espiral se desarrolla en una serie de versiones incrementales, durante las primeras iteraciones, la versión incremental podría ser un modelo en papel o un prototipo. Durante las últimas iteraciones se producen versiones cada vez más completas del sistema diseñado. [Pressman, 2010].





## 2. MARCO TEORICO

## 2. MARCO TEORICO

### 2.1. Software

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

Algunas definiciones de software son:

- El software es conjunto de instrucciones o programas de cómputo que cuando se ejecutan proporcionan las características, función y desempeño buscados, es una estructura de datos que permiten que los programas manipulen en forma adecuada la información [Pressman, 2010].
- Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación. [IEEE. 729].

El software va más allá de los programas de computación en sus distintos estados: código fuente, binario o ejecutable; abarca también su documentación, los datos a procesar e incluso la información de usuario forman parte del software.

## 2.2. Características del Software

El software tiene como principales características:

- El Software se desarrolla o modifica con el intelecto, no se manufactura en el sentido clásico.
- El Software no se desgasta, pero si se deteriora; no es susceptible a los problemas ambientales que hacen que el hardware se desgaste.
- Aunque la industria se mueve hacia la construcción basada en componentes, la mayor parte del software se construye para un uso individualizado.

## 2.3. Ingeniería de Software

La Ingeniería de Software es una disciplina o área de la informática que ofrece métodos o técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

A continuación se mencionan algunas definiciones de Ingeniería de Software

- La Ingeniería de Software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales [Bauer F. 1972].
- La Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantener dichos programas. Así como también el desarrollo de software o producción de software. [Bohem, 1976].
- La Ingeniería de Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software [Zelkovitz, 1978].
- La Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software [IEEE, 1993].

La Ingeniería de Software incluye procesos, métodos y herramientas que permiten:

- Mejorar la calidad de los productos de software.
- Aumentar la productividad y trabajo de los ingenieros del software.
- Facilitar el control del proceso de desarrollo de software.
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.

- Definir una disciplina que garantice la producción y el mantenimiento de los productos software desarrollados en el plazo fijado y dentro del costo estimado.

## **2.4. Ciclo de Vida del Software**

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

### **2.4.1. Ciclo de Vida Clásico**

El método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.

El ciclo de vida clásico llamado también modelo lineal secuencial o modelo en cascada, sugiere un enfoque sistemático, secuencial del desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento. Modelado según el ciclo de ingeniería convencional, el modelo lineal secuencial acompaña a las siguientes actividades:

- **Ingeniería y Modelado de Sistemas/información**

Como el software siempre forma parte de un sistema más grande (o empresa), el trabajo comienza estableciendo requisitos de todos los elementos del sistema y asignando al software algún subgrupo de estos requisitos. Esta visión del sistema es esencial cuando el software se debe interconectar con otros elementos como hardware, personas y bases de datos. La ingeniería y el análisis de sistemas acompañan a los requisitos que se recogen en el nivel del sistema con una pequeña parte de análisis y de diseño. La ingeniería de información acompaña a los requisitos que se recogen en el nivel estratégico de la empresa.

- **Análisis de los Requisitos**

El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del programa o los programas a construirse el analista debe comprender el dominio de la información del software, así como la función requerida, comportamiento, rendimiento e interconexión.

- **Diseño**

El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de un programa: estructura de datos, arquitectura del software, representaciones de interfaz y detalle de procedimental. El proceso de diseño traduce requisitos en una representación del software que se pueda evaluar por calidad antes de que comience la generación de código.

- **Generación de Código**

El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño en una forma detallada, la generación de código se realiza mecánicamente.

- **Pruebas**

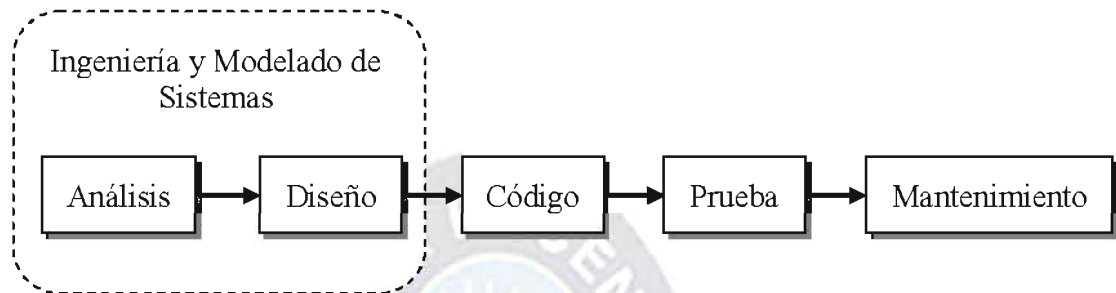
Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias sean comprobadas y en los procesos externos funcionales, es decir, la realización de pruebas para la detección de errores y el sentirse seguro de que la entrada definida produzca resultados reales de acuerdo con los resultados requeridos.

- **Mantenimiento**

El software indudablemente sufrirá cambios después de ser entregados al cliente. Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (por ejemplo se requiere un cambio debido a un dispositivo periférico nuevo), o por que el cliente requiere mejoras funcionales o de rendimiento. El mantenimiento vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo.



El modelo lineal secuencial es el paradigma más antiguo y más extensamente utilizado en la ingeniería de sistemas.



**Figura 2.1.** El Modelo Lineal Secuencial

Fuente: Ingeniería de Software [Pressman, 2010]

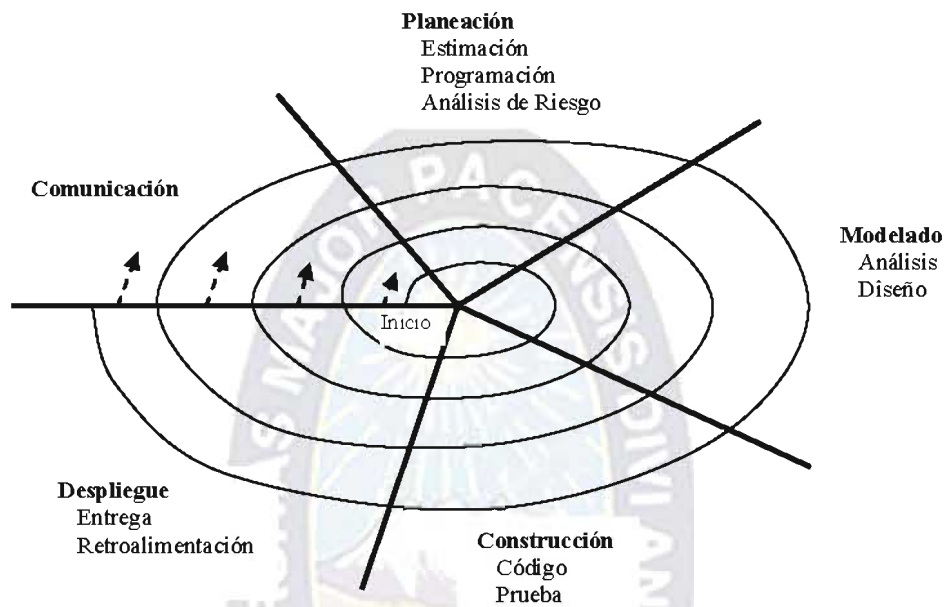
#### **2.4.2. Modelo Espiral**

Propuesto en primer lugar por Barry Boehm [Boe88], el modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo lineal o modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas. Bohem [Boe01a] describe el modelo del modo siguiente:

El modelo de desarrollo espiral es un generador de modelo de proceso impulsado por el riesgo, que se usa para guiar la ingeniería concurrente con participantes múltiples de sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras disminuye su grado de riesgo. La otra es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.

Con el empleo del modelo espiral, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, lo que se entrega puede ser un modelo o prototipo. En las iteraciones posteriores se producen versiones cada vez más completas del sistema.

Un modelo en espiral está dividido en un conjunto de actividades estructurales ilustradas en la siguiente figura:



**Figura 2.2.** El Modelo Espiral

**Fuente:** Ingeniería de Software [Pressman, 2010]

- **Comunicación**

Antes de que comience cualquier trabajo técnico tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto y reunir los requerimientos que ayuden a definir las características y funciones del software.

- **Planeación**

La planeación del proyecto de software, define el trabajo de ingeniería de software al describir las tareas técnicas por realizar, los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.

- **Modelado**

El modelado crea un bosquejo del objeto por hacer a fin de entender el panorama general. Si se requiere refina el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y como resolverlo. Un ingeniero de software crea modelos a fin de entender mejor los requerimientos del software y el diseño que los cumplirá.

- **Construcción**

Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.

- **Despliegue**

El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

Al comenzar el proceso evolutivo, el desarrollador realiza actividades implícitas en un circuito alrededor de la espiral en el sentido horario, partiendo del centro. En cada paso evolutivo se marcan puntos de referencia puntuales; combinaciones de productos del trabajo y condiciones que se encuentran a lo largo de la trayectoria de la espiral.

El primer circuito alrededor de la espiral da como resultado el desarrollo de una especificación del producto; las vueltas sucesivas se usan para desarrollar un prototipo y luego versiones cada vez más sofisticadas del software. Cada paso por la región de planeación da como resultado ajustes en el plan del proyecto. El costo y la programación de actividades se ajustan con base a la retroalimentación obtenida del cliente después de la entrega.

El modelo espiral es un enfoque realista para el desarrollo de sistemas y de software. Como el software evoluciona a medida que el proceso avanza, se puede comprender y reaccionar mejor ante los riesgos en cada nivel de evolución. El modelo en espiral mantiene el enfoque de escalafón sistemático sugerido por el ciclo de vida clásico, pero lo incorpora en una estructura iterativa que refleja el mundo real en una forma más realista.

## 2.5. Ingeniería de Requerimientos

El espectro amplio de tareas y técnicas que llevan a entender los requerimientos se denomina ingeniería de requerimientos. Desde la perspectiva del proceso del software, la ingeniería de requerimientos es una de las acciones más importantes de la ingeniería de software que comienza durante la actividad de comunicación y continúa en la de modelado. Debe adaptarse a las necesidades del proceso, del proyecto, del producto y de las personas que hacen el trabajo.

La ingeniería de requerimientos proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida de que se transforman en un sistema funcional. Incluye siete tareas diferentes:

### 2.5.1. Concepción

En la concepción se establece el entendimiento básico del problema, las personas que quieren una solución, la naturaleza de la solución que se desea, así como la eficacia de la comunicación y colaboración preliminares entre los otros participantes y el equipo de software.

### 2.5.2. Indagación

La indagación pretende determinar cuáles son los objetivos del sistema, qué es lo que va a lograrse, cómo se ajusta el sistema o producto a las necesidades del negocio y como va a usarse el sistema en las operaciones cotidianas.

Esta tarea no es simple, Christel y Kang [Cri92] identificaron cierto número de problemas que se encuentran cuando ocurre la indagación:

- **Problemas de Alcance**

La frontera de los sistemas está mal definida o los clientes o usuarios finales especifican detalles técnicos innecesarios que confunden, más que clarifican, los objetivos generales del sistema.

- **Problemas de Entendimiento**

Los clientes o usuarios no están completamente seguros de lo que se necesita, comprenden mal las capacidades y limitaciones de su ambiente de computación, no entienden todo el dominio del problema, tienen problemas para comunicar las necesidades, omiten información que creen que es obvia, especifican requerimientos que están en conflicto con las necesidades de otros clientes o usuarios, o solicitan requerimientos ambiguos o que no pueden someterse a prueba.

- **Problemas de Volatilidad**

Los requerimientos cambian con el tiempo.

### **2.5.3. Elaboración**

La información obtenida del cliente durante la concepción e indagación se expande y refina durante la elaboración. Esta tarea se centra en desarrollar un modelo refinado de los requerimientos que identifique distintos aspectos de la función del software, su comportamiento e información.

### **2.5.4. Negociación**

En el proceso de negociación se pide a los clientes, usuarios y otros participantes que ordenen sus requerimientos según su prioridad y que después analicen los conflictos. Con el empleo de un enfoque iterativo que da prioridad a los requerimientos, se evalúa su costo y riesgo, y se enfrentan los conflictos internos; algunos requerimientos se eliminan, se combinan o se modifican de modo que cada parte logre cierto grado de satisfacción.

### **2.5.5. Especificación**

Una especificación puede ser un documento escrito, un conjunto de modelos gráficos, un modelo matemático formal, un conjunto de escenarios de uso, un prototipo o cualquier combinación de éstos. Para sistemas grandes, el mejor enfoque puede ser un documento escrito que combine descripciones en un lenguaje natural con modelos gráficos. No obstante, para sistemas pequeños que residan en ambientes bien entendidos, quizá todo lo que se requiera sea escenarios de uso.

### 2.5.6. Validación

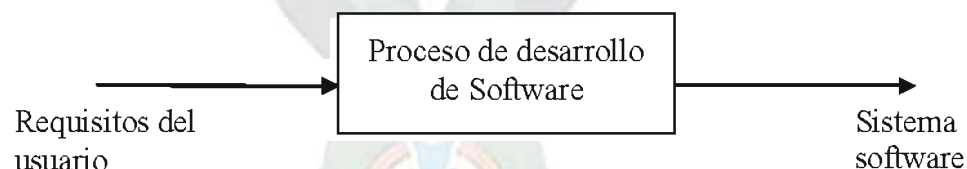
La validación de los requerimientos analiza la especificación a fin de garantizar que todo ellos han sido enunciados sin ambigüedades, que se detectaron y corrigieron las inconsistencias, las omisiones y los errores. El mecanismo principal de validación de los requerimientos es la revisión técnica, en el que se incluye a ingenieros de software, clientes, usuarios y otros participantes, que analizan la especificación en busca de errores de contenido o de interpretación, de aspectos en los que tal vez se requiera hacer aclaraciones, falta de información, inconsistencias y requerimientos en conflictos o irreales.

### 2.5.7. Administración de los Requerimientos

La administración de los requerimientos es el conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar y dar seguimiento a los requerimientos y a sus cambios en cualquier momento del desarrollo del proyecto.

## 2.6. Proceso Unificado de Desarrollo

El Proceso Unificado es un proceso de desarrollo de software, que determina las actividades necesarias para transformar los requisitos de un usuario en un sistema de software.



**Figura 2.3.** Proceso de Desarrollo de Software

**Fuente:** Proceso Unificado de Desarrollo [Jacobson, 2000]

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (Unified Modeling Language, UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado.

El Proceso se define en tres aspectos básicos:

### **2.6.1. Dirigido por Casos de Uso**

Los casos de uso no son solo una herramienta para especificar los requisitos de un sistema. También guían su diseño, implementación y prueba; esto es guían el proceso de desarrollo.

Un sistema software ve la luz para dar servicios a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean.

El termino usuario no solo hace referencia a usuarios humanos sino a otros sistemas. En este sentido, el termino usuario representa alguien o algo que interactúa con el sistema que estamos desarrollando. Ahora bien, una interacción de un usuario con el sistema es considerado como un caso de uso.

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso, el cual describe la funcionalidad de todo el sistema. Puede decirse que una especificación funcional contesta a la pregunta; ¿qué debe hacer el sistema?. La estrategia de casos de uso puede describirse añadiendo al final de esta pregunta: ¿...para cada usuario?. Esto nos fuerza a pensar en términos de importancia para el usuario y no solo en términos de funciones.

### **2.6.2. Centrado en la Arquitectura**

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se reflejan en los casos de uso. Sin embargo, también se ve influida por muchos otros factores, como la plataforma en la que tiene que funcionar el software (hardware, sistema operativo, sistema de gestión

de base de datos, protocolos para comunicación en red), los bloques de de construcción reutilizable que se dispone, consideraciones de implantación, sistemas heredados y requisitos no funcionales. La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado.

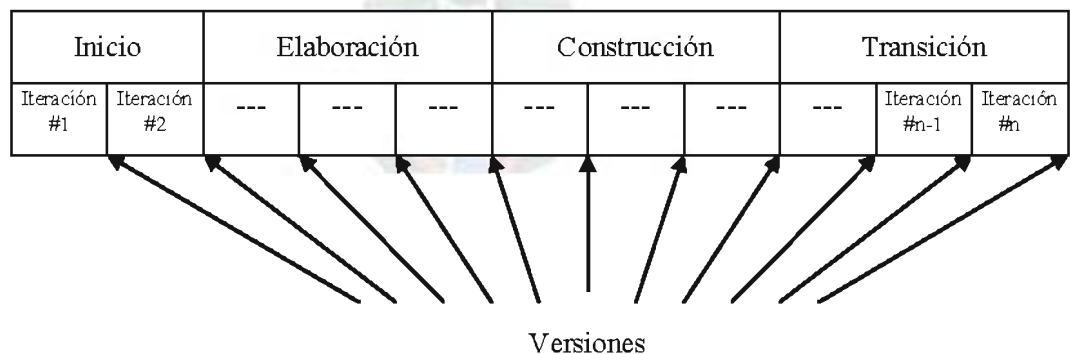
### 2.6.3. Iterativo e Incremental

El desarrollo de un producto software comercial supone de un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

### 2.6.4. La Vida del Proceso Unificado

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes. Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición, cada fase se subdivide a su vez en iteraciones.



**Figura 2.4.** Un Ciclo con sus Fases e Iteraciones

**Fuente:** Proceso Unificado de Desarrollo [Jacobson, 2000]



- **Fase de inicio**

Durante la fase de inicio, se desarrolla una descripción del producto final a partir de una buena idea. Esencialmente, esta fase responde a las siguientes preguntas:

¿Cuáles son las principales funciones del sistema para sus usuarios más importantes?

¿Cómo podría ser la arquitectura del sistema?

¿Cuál es el plan de proyecto y cuánto costará desarrollar el producto?

- **Fase de elaboración**

Durante la fase de elaboración, se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura del sistema. Al final de la fase elaboración, el director del proyecto está en disposición de planificar las actividades y estimar los recursos necesarios para terminar el proyecto.

- **Fase de construcción**

Durante la fase de construcción, se crea el producto. En esta fase, la línea base de la arquitectura crece hasta convertirse en el sistema completo. La descripción evoluciona hasta convertirse en un producto preparado para ser entregado a la comunidad de usuarios. Al final de esta fase, el producto contiene todos los casos de uso que la dirección y el cliente han acordado para el desarrollo de esta versión. Sin embargo, puede que no esté completamente libre de defectos. Muchos de estos defectos se descubrirán y solucionarán durante la fase de transición. La pregunta decisiva es: ¿cubre el producto las necesidades de algunos usuarios de manera suficiente como para hacer una primera entrega?.

- **Fase de transición**

La fase de transición cubre el periodo durante el cual el producto se convierte en versión beta. En la versión beta un número reducido de usuarios con experiencia prueba el producto e informa de defectos y deficiencias. Los desarrolladores corrigen los problemas e incorporan algunas de las mejoras sugeridas en una versión general dirigida a la totalidad de la comunidad de usuarios.

### 2.6.5. Flujos de Trabajo

En el Proceso Unificado cada fase se divide normalmente en iteraciones o mini proyectos. Una iteración típica pasa por cinco flujos de trabajo que son: requisitos, análisis, diseño, implementación y prueba.

Los desarrolladores comienzan capturando los requisitos del cliente en la forma de casos de uso. Después analizan y diseñan el sistema para cumplir los casos de uso, creando en primer lugar un modelo de análisis, después uno de diseño y después otro de implementación, el cual incluye todo el código, es decir los componentes. Por último los desarrolladores preparan un modelo de prueba que les permite verificar que el sistema proporciona la funcionalidad descrita en los casos de uso.

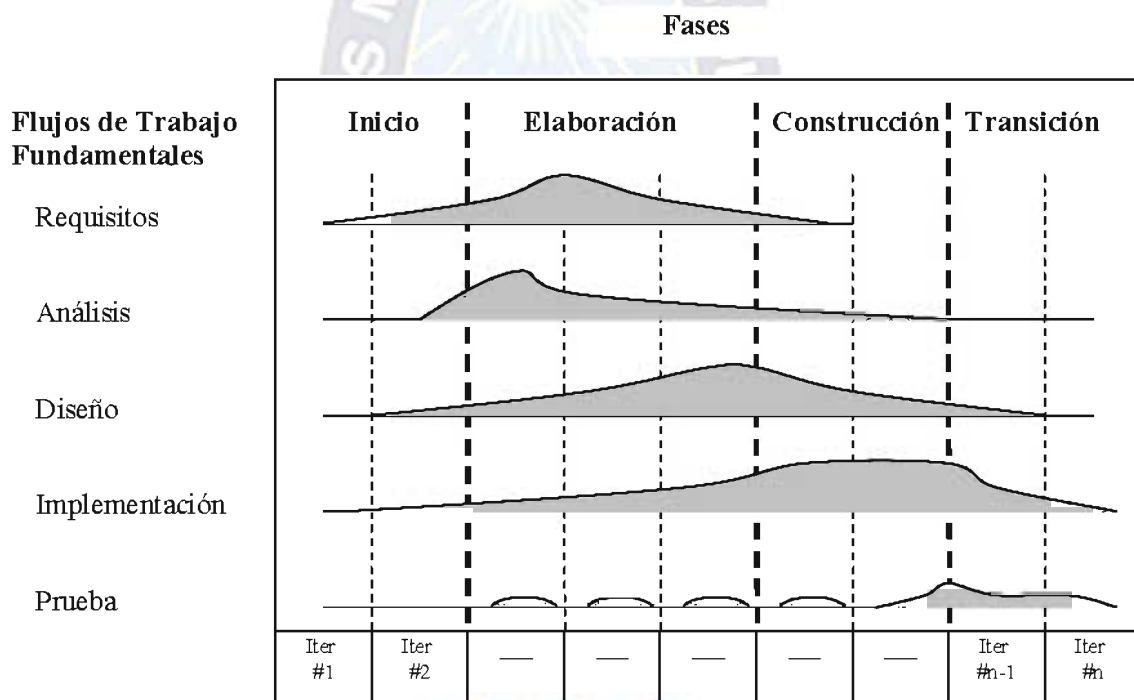


Figura 2.5. Flujos de Trabajo y Fases

Fuente: Proceso Unificado de Desarrollo [Jacobson, 2000]

### 2.7. UML

El lenguaje unificado de modelado o UML (Unified Modeling Language) es un lenguaje de modelado, y no un método, permite visualizar, especificar, construir y documentar sistemas de software. [Fowler, 1997]

### 2.7.1. Casos de Uso

Un caso de uso es, en esencia, una interacción típica entre un usuario y un sistema de cómputo. El modelo de casos de uso especifica la funcionalidad que el sistema ha de ofrecer desde la perspectiva de los usuarios y lo que el sistema ha de realizar para satisfacer las peticiones de estos usuarios. El diagrama de casos de uso visualiza el comportamiento de un sistema.

Este modelo utiliza tres elementos básicos:

- **Actores**

El actor es un tipo o categoría de usuarios; cuando un usuario realiza una tarea con el sistema, actúa como una ocurrencia de ese tipo de usuarios. También, un usuario puede desempeñar los papeles de diferentes actores. De este modo, un actor define los diferentes papeles que un usuario puede desempeñar.

El conjunto de actores representa a todo aquello que necesita intercambiar información con el sistema que se está desarrollando. Los actores pueden modelar a personas, pero también se pueden corresponder con otros sistemas que se comunican con el que sea objeto de desarrollo. El aspecto esencial es que los actores constituyen todo aquello que es externo al sistema que se está desarrollando.

- **Casos de uso**

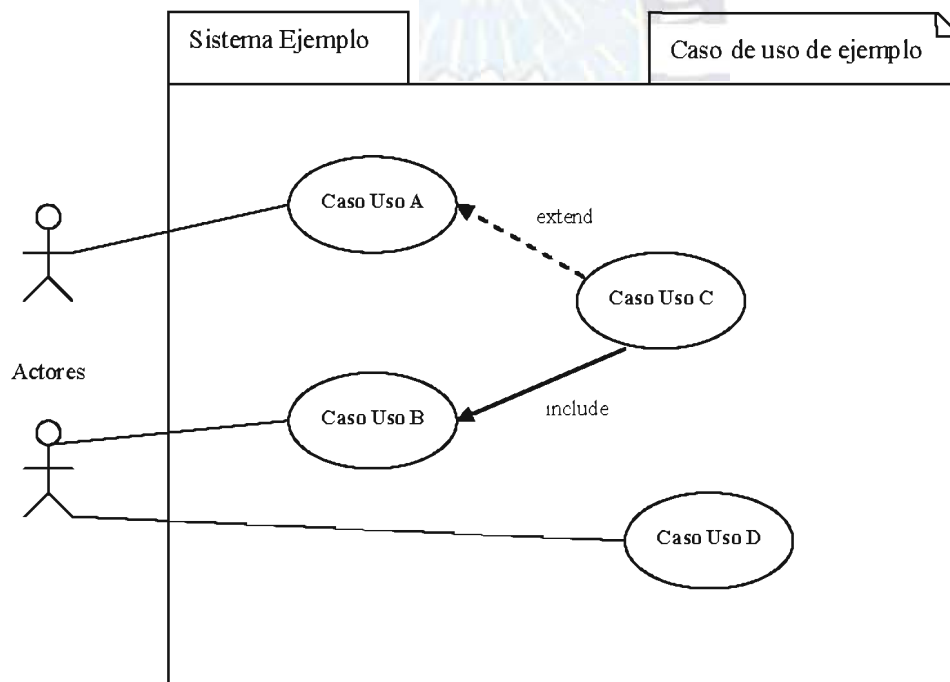
Un caso de uso puede ser entendido como la secuencia de transacciones que se realizan en un dialogo con el sistema, y que se encuentran relacionadas con su comportamiento. Cada caso de uso constituye una secuencia completa de mensajes, y que especifica la secuencia de interacción que tiene lugar entre un actor y el sistema.

Por lo tanto, se puede decir que un caso de uso es una secuencia de transacciones relacionadas, ejecutadas por uno o más actores y el sistema en un dialogo determinado. La colección de todos los casos de uso relacionados con un sistema especifica todas las maneras en que se puede utilizar el sistema.

- **Relaciones**

Para identificar la comunicación existente entre los actores, los casos de uso y actores – caso de uso se utilizan varios tipos de relaciones:

- La relación de asociación entre un actor y un caso de uso se utiliza para señalar la comunicación
- La relación de extensión (extend) entre casos de uso se utiliza para factorizar las variantes sobre la secuencia básica de un caso de uso en nuevos casos de uso que extienden los flujos principales.
- La relación inclusión (incluye) entre casos de uso se utiliza para señalar que un caso de uso incorpora el comportamiento de otro caso de uso como parte de su propio comportamiento.



**Figura 2.6.** Diagrama de Casos de Uso

Fuente: Análisis y Diseño Orientado a Objetos [Seco, 2003]

### 2.7.2. Diagrama de Clases

El diagrama de clases modela la vista estática del sistema, recoge los conceptos del dominio de la aplicación.

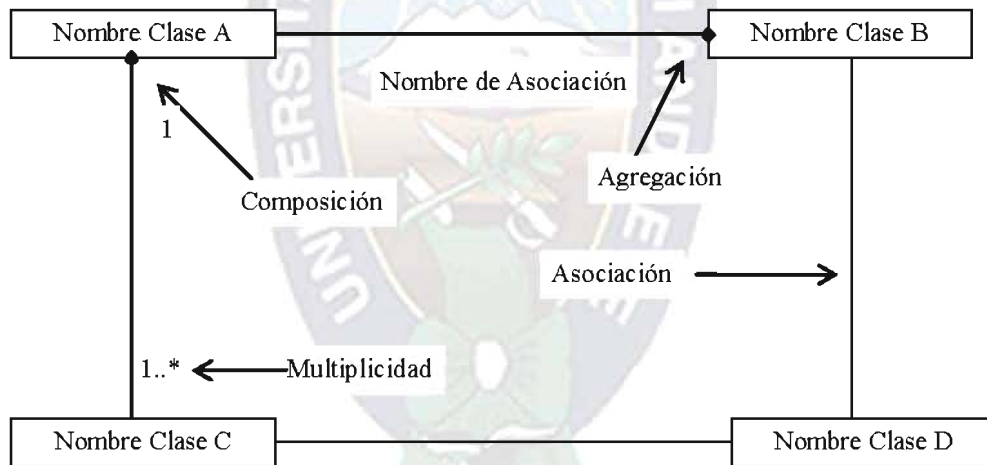
Los elementos principales de un diagrama de clases son:

- **Clases**

Una clase es una descripción de un conjunto de objetos con las mismas propiedades (atributos), el mismo comportamiento (operaciones), las mismas relaciones con otros objetos y la misma semántica. Una clase puede representar un concepto del mundo real (modelo de análisis) o un concepto de implementación (modelo del diseño) del sistema que se está modelando.

- **Relaciones**

Una relación es una abstracción de un conjunto de interrelaciones semánticas puntuales que se dan sistemáticamente entre diferentes tipos de objetos.



**Figura 2.7.** Diagrama de Clases

**Fuente:** Análisis y Diseño Orientado a Objetos [Seco, 2003]

### 2.7.3. Diagrama de Transición de Estados

Los diagramas de transición de estados son una técnica para describir el comportamiento de un sistema. Describen todos los estados posibles en los que se encontrar un objeto particular y la manera que cambia el estado del objeto.

Un diagrama de transición de estados tiene los siguientes elementos:

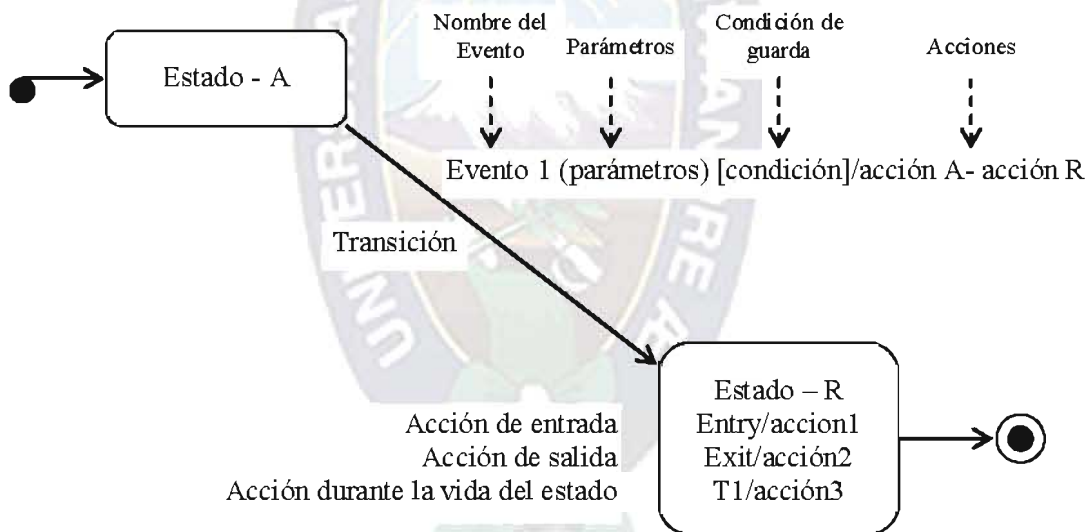
- **Estado**

Un estado es el periodo de tiempo que tiene lugar entre la recepción de dos eventos consecutivos por parte de una clase del sistema. Durante el tiempo que dura un estado pueden ocurrir varias cosas:

- Una acción a la entrada del estado.
- Una acción durante el tiempo que dura el estado.
- Una acción al salir del estado.

- **Transición**

Una transición tiene lugar entre dos estados en los que se puede encontrar una clase. Puede tener una acción y/o una condición de guarda asociada a la transición. Además puede disparar un evento. [Seco, 2003]



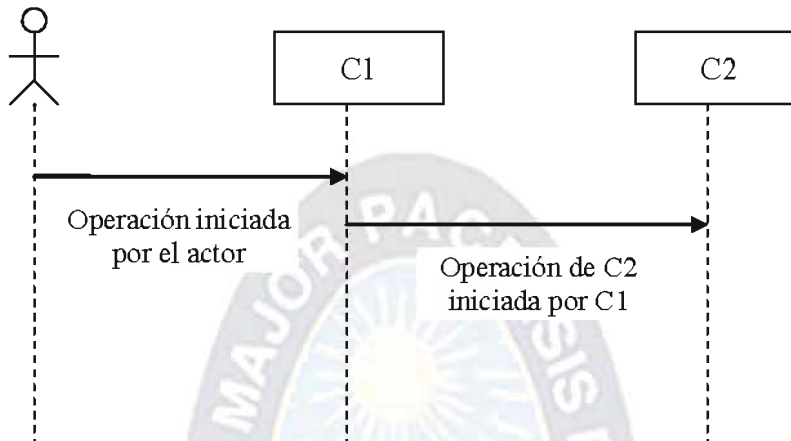
**Figura 2.8.** Diagrama de Transición de Estados

Fuente: Análisis y Diseño Orientado a Objetos [Seco, 2003]

#### 2.7.4. Diagramas de Secuencia

Un diagrama de secuencia representa la interacción entre clases del modelo de estructuras estáticas, ordenada temporalmente.

Normalmente cada caso de uso tiene asociado varios diagramas de secuencia, uno que representa el curso típico de funcionamiento del caso de uso y uno o más por cada posible ejecución alternativa del caso de uso, es decir, si se producen errores, etc.



**Figura 2.9.** Diagrama de Secuencia

**Fuente:** Análisis y Diseño Orientado a Objetos [Seco, 2003]

## 2.8. Base de Datos

Una base de datos es un conjunto de información estructurada en registros y almacenada en un soporte electrónico legible desde un ordenador. Cada registro constituye una unidad autónoma de información que puede estar a su vez estructurada en diferentes campos o tipos de datos que se recogen en dicha base de datos.

### 2.8.1. Base de Datos Relacional

Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura, este conjunto de esquemas relacionales permite almacenar la información con un mínimo de redundancia y facilita la recuperación de la información. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.

### 2.8.2. Gestor de Base de Datos

Es un conjunto de herramientas que suministra a todos (administradores, analistas, programadores, usuarios), los medios necesarios para describir, recuperar y manipular

los datos almacenados en la base de datos, manteniendo la seguridad, integridad y confidencialidad de los mismos.

Ofrece a la empresa un control centralizado de su información evitando la redundancia de los datos, mejorando los mecanismos de seguridad y privacidad, asegurando la independencia de los programas con respecto a los datos, manteniendo la integridad de los datos, mejorando la eficacia de acceso a los datos. [monog BD]

### **2.8.3. Gestor de Base de Datos MySql**

MySql es un gestor de base de datos accesible de forma gratuita y está disponible para las principales plataformas del mercado, es sencillo de usar e increíblemente rápido. Al margen de algunas herramientas y pequeños programas para diferentes usos, básicamente se compone de un programa que actúa de servidor y que atiende las peticiones de los clientes, es uno de los motores de base de datos más usados en Internet.

Las características principales de MySql son:

- **Interioridades y Portabilidad**
  - Escrito en C y en C++.
  - Funciona en diferentes plataformas.
  - APIs disponibles para C, C++, Java, Perl, PHP, Python, Ruby y Tcl.
  - Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
  - Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
  - Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfaz SQL para una base de datos propia.
  - Un sistema de reserva de memoria muy rápido basado en threads.
  - Joins muy rápidos usando un multi-join de un paso optimizado.
  - Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
  - El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linqueado) en aplicaciones autónomas. Dichas



aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.

- **Seguridad**

- Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está cifrado cuando se conecta con un servidor.

- **Escalabilidad y límites**

- Soporte a grandes bases de datos que contienen 50 millones de registros. También se conoce a usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2).

- **Conectividad**

- Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
- La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible.
- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.

- **Localización**

- El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.

- Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son case-insensitive.
  - La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL.
- **Clientes y Herramientas**
    - MySql server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente mysqlcheck. MySQL también incluye myisamchk una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas MyISAM.
    - Todos los programas MySQL pueden invocarse con las opciones – help 0 -? para obtener asistencia en línea. [wiki MySql]

## 2.9. PHP

PHP es un lenguaje de scripting que permite la generación dinámica de contenidos en un servidor. Su nombre oficial es PHP HyperText Preprocessor. Entre sus principales características se pueden destacar su potencia, alto rendimiento.

PHP es un lenguaje de programación que contiene muchos conceptos de C, Perl y Java. Su sintaxis es muy similar a la de estos lenguajes, haciendo muy sencillo su aprendizaje.

### 2.9.1. Características de PHP

Entre sus principales características se pueden destacar las siguientes:

- Es un potente y robusto lenguaje de programación embebido en documentos HTML.
- Dispone de librerías de conexión con la gran mayoría de los sistemas de gestión de base de datos para el almacenamiento de información permanente en el servidor.
- Proporciona soporte a múltiples protocolos de comunicación en Internet (HTTP, IMAP, FTP, etc.).

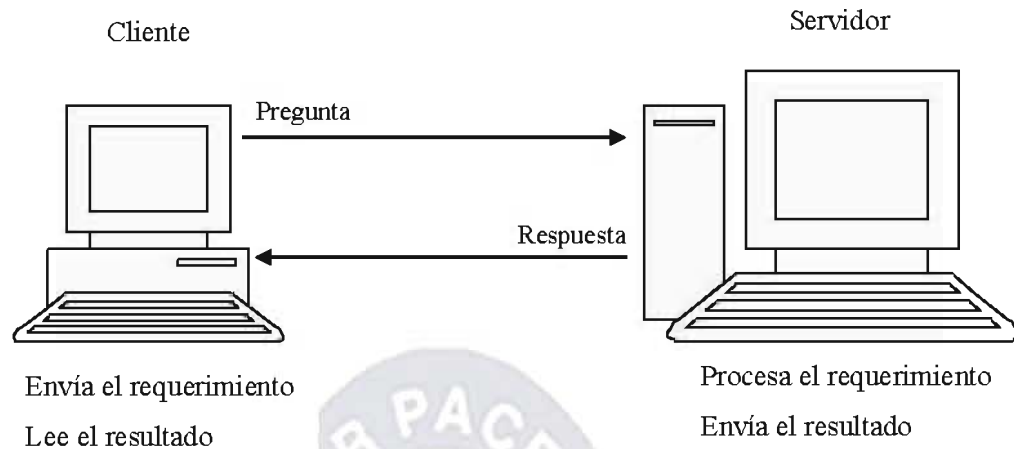
- Código fuente abierto; el código del intérprete está accesible para permitir posibles mejoras o sugerencias acerca de su desarrollo (PHP ha sido escrito en lenguaje C).
- Gratuito; no es necesario realizar ningún desembolso económico para desarrollar sistemas de información empleando este versátil lenguaje.
- Eficiente; PHP consume muy pocos recursos en el servidor.
- Alta velocidad de desarrollo; PHP permite desarrollar interesantes aplicaciones, pues proporciona gran cantidad de librerías muy útiles y bien documentadas que ahorran mucho trabajo al programador.
- También dispone de facilidades para el procesamiento de ficheros, funciones de tratamiento de textos, generación dinámica de imágenes, tratamiento de documentos XML. [Gil 2001]

## **2.10.Arquitectura Cliente/Servidor**

La arquitectura Cliente/Servidor empezó a ser aceptado a finales de los ochentas. Su funcionamiento es sencillo: se tiene una máquina cliente, que requiere un servicio de una máquina servidor, y éste realiza la función para la que está programado (nótese que no tienen que tratarse de máquinas diferentes; es decir, una computadora por sí sola puede ser ambos cliente y servidor dependiendo del software de configuración).

Desde el punto de vista funcional, se puede definir al modelo Cliente/Servidor como una arquitectura distribuida que permite a los usuarios finales obtener acceso a la información en forma transparente aún en entornos multiplataforma.

En el modelo Cliente/Servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio).



**Figura 2.10.** Modelo Cliente/Servidor

**Fuente:** Administración de Base de Datos

En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. La idea es tratar a una computadora como un instrumento, que por sí sola pueda realizar muchas tareas, pero con la consideración de que realice aquellas que son más adecuadas a sus características. Si esto se aplica tanto a clientes como servidores se entiende que la forma más estándar de aplicación y uso de sistemas Cliente/Servidor es mediante la explotación de las PC's a través de interfaces gráficas de usuario; mientras que la administración de datos y su seguridad e integridad se deja a cargo de computadoras centrales tipo mainframe.

Usualmente la mayoría del trabajo pesado se hace en el procesador llamado servidor y el o los procesos cliente sólo se ocupan de la interacción con el usuario (aunque esto puede variar).

### 2.10.1. Cliente

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor. El cliente normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

### **2.10.2. Servidor**

Es cualquier recurso de cómputo encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

### **2.10.3. Características de la Arquitectura Cliente/Servidor**

Las características básicas de una arquitectura Cliente/Servidor son:

- Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- Las tareas del cliente y del servidor tienen diferentes requerimientos en cuanto a recursos de cómputo como velocidad del procesador, memoria, velocidad y capacidades del disco dispositivos de entrada y salida.

- Se establece una relación entre procesos distintos, los cuales pueden ser ejecutados en la misma máquina o en máquinas diferentes distribuidas a lo largo de la red.
- Existe una clara distinción de funciones basada en el concepto de "servicio", que se establece entre clientes y servidores.
- La relación establecida puede ser de muchos a uno, en la que un servidor puede dar servicio a muchos clientes, regulando su acceso a recursos compartidos.
- Los clientes corresponden a procesos activos en cuanto a que son éstos los que hacen peticiones de servicios a los servidores. Estos últimos tienen un carácter pasivo ya que esperan las peticiones de los clientes.
- No existe otra relación entre clientes y servidores que no sea la que se establece a través del intercambio de mensajes entre ambos. El mensaje es el mecanismo para la petición y entrega de solicitudes de servicio.
- El ambiente es heterogéneo. La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma. Precisamente una de las principales ventajas de esta arquitectura es la posibilidad de conectar clientes y servidores independientemente de sus plataformas.
- El concepto de escalabilidad tanto horizontal como vertical es aplicable a cualquier sistema Cliente/Servidor. La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar significativamente el rendimiento. La escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

#### **2.10.4. Ventajas de la Arquitectura Cliente/Servidor**

Entre las principales ventajas de la arquitectura Cliente/Servidor están:

- Uno de los aspectos que más ha promovido el uso de sistemas Cliente/Servidor, es la existencia de plataformas de hardware cada vez más baratas. Esta constituye a su vez una de las más palpables ventajas de este esquema, la

posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.

- El esquema Cliente/Servidor facilita la integración entre sistemas diferentes y comparte información permitiendo, por ejemplo que las máquinas ya existentes puedan ser utilizadas pero utilizando interfaces mas amigables al usuario. De esta manera, podemos integrar PCs con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operativo.
- Al favorecer el uso de interfaces gráficas interactivas, los sistemas construidos bajo este esquema tienen mayor interacción y más intuitiva con el usuario. En el uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja, con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- Una ventaja adicional del uso del esquema Cliente/Servidor es que es más rápido el mantenimiento y el desarrollo de aplicaciones, pues se pueden emplear las herramientas existentes (por ejemplo los servidores de SQL o las herramientas de más bajo nivel como los sockets o el RPC).
- La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.

#### **2.10.5. Desventajas de la Arquitectura Cliente/Servidor**

Entre las principales desventajas de la arquitectura Cliente/Servidor están:

- El mantenimiento de los sistemas es más difícil pues implica la interacción de diferentes partes de hardware y de software, distribuidas por distintos proveedores, lo cual dificulta el diagnóstico de fallas.
- Se cuenta con muy escasas herramientas para la administración y ajuste del desempeño de los sistemas.
- Es importante que los clientes y los servidores utilicen el mismo mecanismo (por ejemplo sockets o RPC), lo cual implica que se deben tener mecanismos generales que existan en diferentes plataformas.
- Además, hay que tener estrategias para el manejo de errores y para mantener la consistencia de los datos.
- La seguridad de un esquema Cliente/Servidor es otra preocupación importante. Por ejemplo, se deben hacer verificaciones en el cliente y en el servidor.
- El desempeño es otro de los aspectos que se deben tener en cuenta en el esquema Cliente/Servidor. Problemas de este estilo pueden presentarse por congestión en la red, dificultad de tráfico de datos, etc. [capítulo5]

## 2.11. Planificación

La planificación es la más básica de las funciones administrativas, comprende en seleccionar misiones y objetivos y las acciones para alcanzarlos; se requiere tomar decisiones, es decir, seleccionar entre cursos de acción futuros alternativos. Por lo tanto, los planes proporcionan un enfoque racional a objetivos preseleccionados.

La planificación salva la brecha existente entre dónde se está y a dónde se quiere llegar. Hace posible que ocurran cosas que de lo contrario no ocurrirían. Aunque es muy raro que se puede predecir con exactitud el futuro y aunque factores fuera de nuestro control puedan interferir con los planes mejor concebidos, a menos de que se elaboren planes que se estarán dejando los acontecimientos a la casualidad. La planeación es un proceso intelectualmente exigente; se requiere determinar, en forma



consciente, cursos de acción y basar las decisiones en los objetivos, el conocimiento y estimaciones consideradas.

En el proceso de planificación se puede definir los siguientes pasos:

- **Establecer objetivos o metas**

Los objetivos especifican los resultados esperados y señalan lo que se tiene que hacer en definitiva, a qué habrá de darse prioridad y que tendrá que lograrse.

- **Desarrollo de premisas**

Se deben establecer, difundir y obtener consenso para utilizar premisas tales como pronósticos, políticas básicas y planes existentes en la compañía. Son suposiciones sobre el medio ambiente en el cual se llevará a cabo el plan.

- **Determinación de cursos alternativos**

En este paso se deben buscar y examinar cursos alternativos de acción, en especial aquellos que no se manifiestan claramente.

- **Evaluación de cursos alternativos**

Después de buscar cursos alternativos y examinar sus puntos fuertes y débiles el siguiente paso es evaluarlos, preponderando los que se ajusten a las premisas y a los objetivos.

- **Seleccionar un curso**

Este es el punto en el que se adopta el plan, el punto real de toma de decisiones.

- **Elaboración de planes de apoyo**

Cuando se toma la decisión, pocas veces la planeación está completa; por lo tanto, en casi todos los casos se requiere de planes derivados para respaldar el plan básico.

## 2.12. Control

El control es la medición y corrección de las actividades de los subordinados para asegurar que los acontecimientos estén de acuerdo con los planes. Mide el

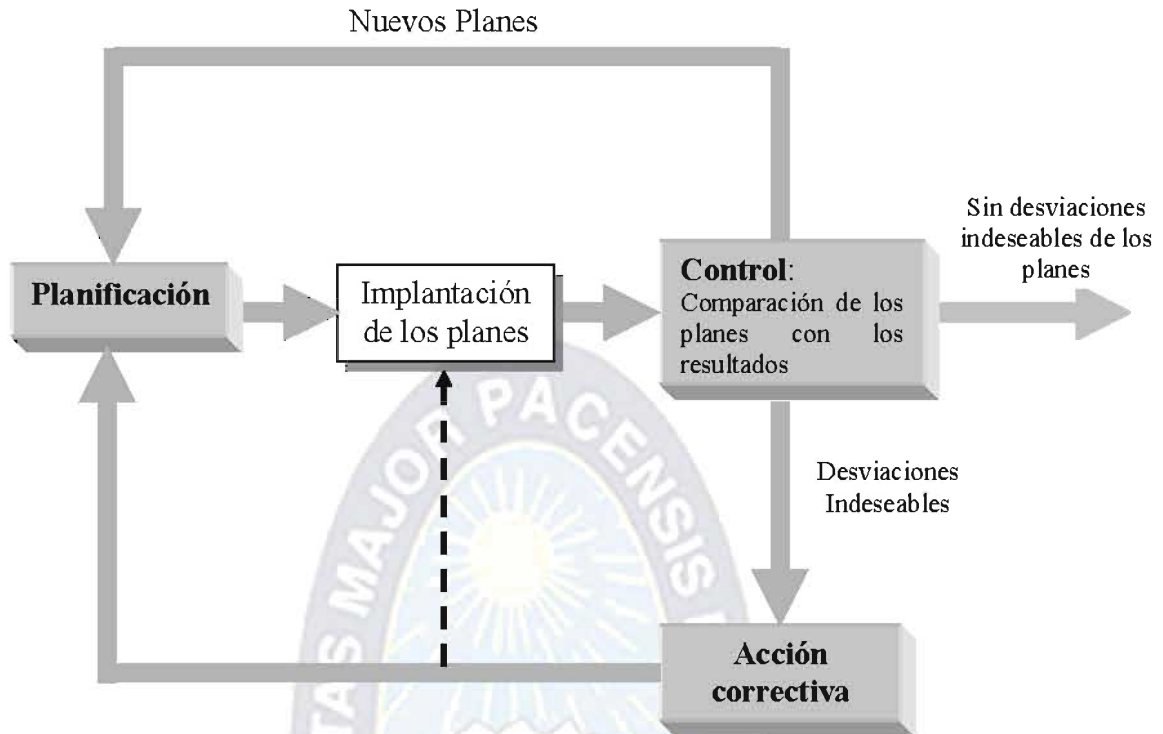
desempeño contra las metas y planes, muestra cuando existen desviaciones negativas y al poner en marcha acciones para corregir las desviaciones, ayuda a asegurar el cumplimiento de los planes.

Las actividades de control, por lo general, se relacionan con la medición del cumplimiento de lo planificado. Algunos medios de control, como el presupuesto de gastos, los registros de inspección y el registro de las horas de trabajo perdidas, son generalmente bien conocidos. Cada uno mide y muestra si los planes están funcionando. Si persisten las desviaciones se señala la corrección. El forzar los acontecimientos para que estén de acuerdo con los planes significa identificar a las personas que tienen la responsabilidad de resultados que son diferentes de la acción planeada y después llevar a cabo los pasos necesarios para mejorar el desempeño. Por lo tanto, los resultados se controlan verificando lo que hace la gente. [Koontz, 1991].

### **2.13. Relación Entre Planificación y Control**

La planeación y el control son inseparables. Cualquier intento de ejercer control sin planes no tiene significado, puesto que no existe alguna forma en que las personas puedan saber si están dirigiéndose hacia donde quieren ir (el resultado de la tarea de control), a menos de que primero se conozcan a dónde quieren ir (parte de la tarea de la planificación). Por consiguiente, los planes proporcionan los estándares de control.

La planificación y el control se tienen que contemplar como las hojas de unas tijeras; éstas no pueden funcionar a menos que existan dos hojas. Sin objetivos y planes no es posible el control, porque el desempeño se tiene que comparar con algunos criterios establecidos.



**Figura 2.11.** Relación entre la Planificación y el Control  
**Fuente:** Elementos de Administración [Koontz, 1991]

## 2.14. Teoría de Inventarios

Para satisfacer la demanda a tiempo, las compañías suelen tener a mano los materiales necesarios transformarlos en productos que esperan vender. El propósito de la teoría de inventarios es determinar las reglas que puede usar la administración para minimizar los costos asociados con mantener el inventario y satisfacer la demanda de los clientes.

### 2.14.1. Modelo de Lote Económico de Pedido (EOQ).

El modelo de lote económico de pedido (EOQ, economic order quantity) se puede usar para tomar decisiones de inventarios óptimas cuando la demanda es determinística (conocida por adelantado).

Para que se cumpla el modelo básico de EOQ, se requieren ciertas suposiciones:

- a) La demanda es determinística y ocurre a una tasa constante.

- b) Si se hace un pedido de cualquier tamaño (por ejemplo, q unidades), se incurre en un costo K de pedido y organización.
- c) El tiempo de espera de cada pedido es cero.
- d) No se permite escasez.
- e) El costo por unidad – año de inventario de reserva es h.
- f) La ecuación para determinar el lote económico de pedido es:

$$q^* = \left( \frac{2KD}{h} \right)^{1/2}$$

Donde:

**q\***: la cantidad de unidades por pedido

**K**: costo de hacer un pedido

**D**: la demanda por año (número de unidades pedidas por año)

**h**: costo de mantener el inventario por unidad – año.

Para determinar el tiempo entre pedidos se tiene:

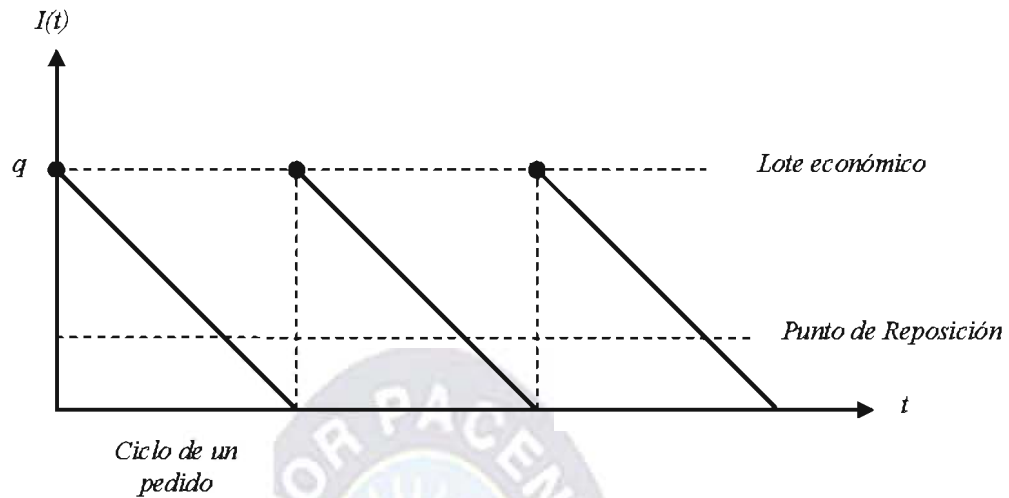
$$t = \frac{q^*}{D}$$

Donde:

**t**: es el tiempo entre pedidos.

Ahora cuando el plazo de entrega L sea mayor a cero sin cambios en los costos anuales de mantener el inventario y el costo por pedido se tiene que determinar el punto de reposición para evitar que haya escasez.

Cada pedido se debe hacer a un nivel de inventario que asegure que cuando llega el pedido, el nivel de inventario sea igual a cero. El nivel de inventario al que se debe hacer un pedido es el punto de reposición.



**Figura 2.12.** Modelo de Lote Económico de Pedido (EOQ)

**Fuente:** Investigación de Operaciones [Wayne, 2005]

## 2.15. Algoritmos Voraces

Hay muchos problemas en los que se pretende obtener un subconjunto de  $n$  elementos que satisfaga ciertas restricciones y que optimice alguna medida. Se supone que un problema de esta clase tiene al menos una solución. Puede haber varias soluciones óptimas, en cuyo caso no importa cual se elija. A menudo, el problema incluye restricciones adicionales que limitan el número posible de soluciones.

Normalmente, estos problemas no se intentan resolver “de golpe”, encontrando de una sola vez la solución completa y óptima. Es más frecuente que el subconjunto de la solución se vaya formando paso a paso, analizando durante cada etapa qué elemento conviene añadir a la solución parcial ya existente.

Un algoritmo voraz sigue el esquema anterior, pero con la fortuna de que cada vez que se añade un elemento a la solución se tiene la certeza de haber realizado la mejor elección posible. Esta característica hace que aunque el análisis del problema sea arduo, la solución voraz siempre resulte sencilla. La única complicación es comprobar que se siguen satisfaciendo las restricciones del problema.

Por lo que se ha descrito del esquema voraz, éste es un proceso repetitivo sencillo que trata sucesivamente los diferentes elementos del problema. Para facilitar la descripción

de este proceso, puede llamarse candidato al elemento tratado en cada paso. Inicialmente, el conjunto de candidatos que forman la solución está vacío. En cada paso se intenta añadir el mejor de los candidatos restantes a dicha solución parcial. Si este conjunto sigue siendo válido, es decir, si satisface las restricciones del problema y, por tanto, permite formar una solución del problema, el candidato se incorpora definitivamente. Al contrario si dicho conjunto no es válido, se desecha el candidato. Si el algoritmo voraz se ha diseñado correctamente, la primera solución completa encontrada es óptima. Por tanto, la dificultad principal al diseñar un algoritmo voraz reside en encontrar un criterio de selección que garantice que la solución sea la óptima.

Según esta descripción, el problema parte de:

- Una Función Objetivo queda el valor de una solución. Obviamente, ésta es la función (solución) por optimizar.
- Un conjunto de restricciones sobre el valor de los datos de entrada y sobre la solución final del problema.

A su vez, la solución consta de:

- Un conjunto de candidatos.
- Una función de selección que en cada momento determine qué candidato de los aún no usados parece ser el mejor.
- Una función que determine si cierto conjunto de candidatos es válido; es decir, si permite formar alguna solución del problema.
- Una función que determine si cierto conjunto de candidatos constituye una solución del problema (no necesariamente la mejor).

## **2.16. Calidad del Software**

Una de las principales fases dentro de la elaboración de un proyecto es el aseguramiento de la calidad del software, es decir, un modelo sistemático y planeado de todas las acciones necesarias para proveer la confianza adecuada, según los requerimientos técnicos establecidos, de cada producto e ítem del proyecto.

- **Definiciones de Calidad de Software.**
  - “La calidad del software es el grado con el que un sistema, componente o un proceso cumple con los requerimiento especificados, y las necesidades o expectativas del cliente o usuario”. (IEEE Std. 610 – 1990).
  - “Conjunto de características de una entidad (producto o servicio) que le confieren su aptitud para satisfacer necesidades expresadas o implícitas”. (ISO 8402 – UNE 66 – 001 -92).

#### **2.16.1. Factores de la Calidad ISO 9126.**

El estándar ISO 9126 se desarrollo con la intención de identificar los atributos clave del software. Este sistema identifica seis atributos de la calidad.

- **Funcionalidad**

Grado con el que el software satisface las necesidades implícitas y explícitas, según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.
- **Confiabilidad**

Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallos y recuperación.
- **Usabilidad**

Grado en el que el software es fácil de usar, según lo indican los siguientes atributos: entendible, aprendible y operable.
- **Eficiencia**

Grado en que el software emplea óptimamente los recursos del sistema, según lo indican los atributos siguientes: comportamiento del tiempo y de los recursos.
- **Facilidad de Recibir Mantenimiento**

Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable, susceptible de someterse a pruebas.

- **Portabilidad**

Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

La ISO 9126 no necesariamente conduce a una medición directa. Sin embargo, proporcionan una base útil para hacer mediciones indirectas y una lista de comprobación excelente para evaluar la calidad del sistema.

## 2.17. Seguridad

La seguridad informática está concebida para proteger los activos informáticos, entre los que se encuentran:

- **La Información Contenida**

La información se ha convertido en uno de los elementos más importantes dentro de una organización. La seguridad informática debe ser administrada según los criterios establecidos por los administradores y supervisores, evitando que usuarios externos y no autorizados puedan acceder a ella sin autorización. De lo contrario la organización corre el riesgo de que la información sea utilizada maliciosamente para obtener ventajas de ella o que sea manipulada, ocasionando lecturas erradas o incompletas de la misma. Otra función de la seguridad informática en esta área es la de asegurar el acceso a la información en el momento oportuno, incluyendo respaldos de la misma en caso de que esta sufra daños o pérdida producto de accidentes, atentados o desastres.

- **La Infraestructura Computacional**

Una parte fundamental para el almacenamiento y gestión de la información, así como para el funcionamiento mismo de la organización. La función de la seguridad informática en esta área es velar que los equipos funcionen adecuadamente y prever en caso de falla planes de robos, incendios, boicot, desastres naturales, fallas en el suministro eléctrico y cualquier otro factor que atente contra la infraestructura informática.

- **Los Usuarios**

Son las personas que utilizan la estructura tecnológica, zona de comunicaciones y que gestionan la información. La seguridad informática debe establecer normas que



minimicen los riesgos a la información o infraestructura informática. Estas normas incluyen horarios de funcionamiento, restricciones a ciertos lugares, autorizaciones, denegaciones, perfiles de usuario, planes de emergencia, protocolos y todo lo necesario que permita un buen nivel de seguridad informática minimizando el impacto en el desempeño de los funcionarios y de la organización en general y como principal contribuyente al uso de programas realizados por programadores.

Los aspectos mencionados pueden ser viabilizados a través de la seguridad de acceso y de la autenticación de los usuarios [wiki segur].



### **3. MARCO APLICATIVO**



### 3. MARCO APLICATIVO

#### 3.1. Antecedentes

La fábrica “Intiwara” se encuentra ubicada en la ciudad de La Paz, en la zona de Sopocachi, tiene como principal actividad la confección de chompas de alpaca, para su comercialización principalmente en el ámbito internacional, teniendo también una tienda en la plaza San Francisco en el centro de nuestra ciudad. Para poder satisfacer los pedidos de los clientes se subcontrata talleres artesanales en las temporadas altas de ventas.

- **Objetivo**

El principal objetivo es la generación de recursos económicos para poder cumplir con las actividades propuestas y a partir de esto generar ganancias para los propietarios como retribución de los recursos invertidos por estos.

- **Misión**

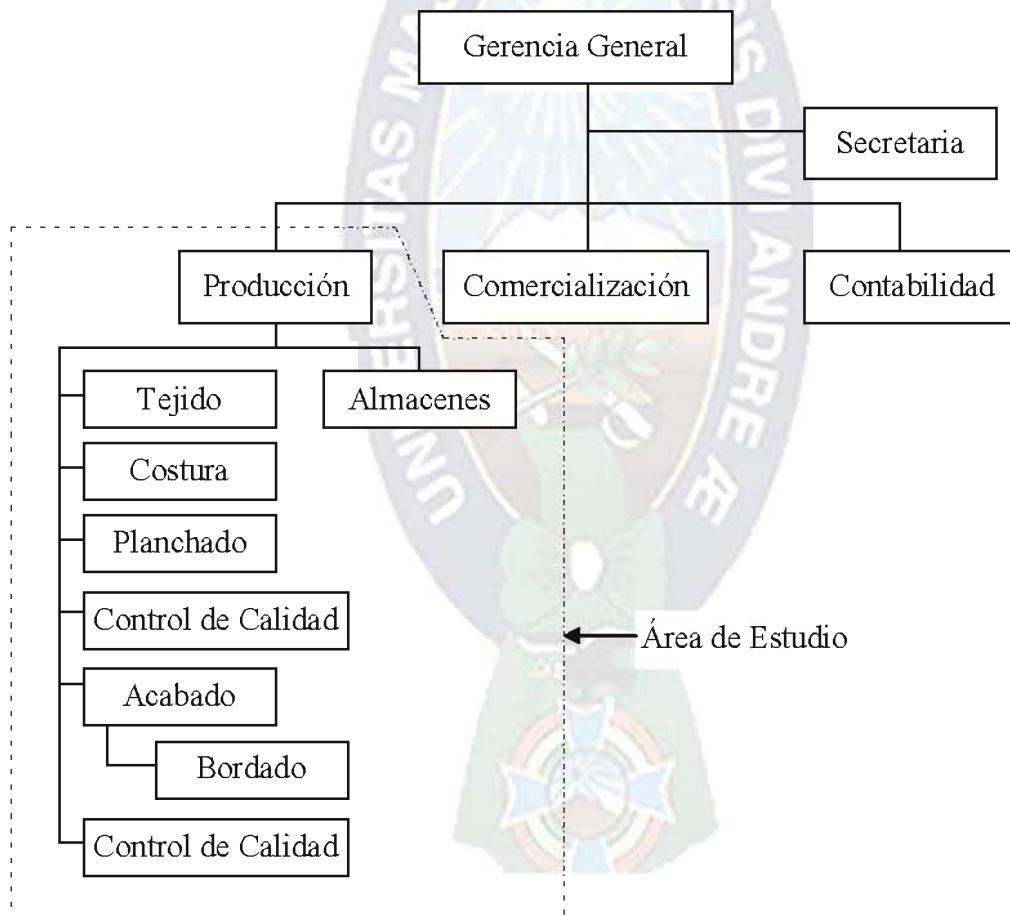
La misión es poder consolidarse como una fábrica de renombre nacional e internacional, capaz de cumplir con los requerimientos de los clientes actuales y la ampliación de la cartera de clientes como consecuencia de la confección de productos de alta calidad.

- **Visión**

Proyectar una imagen de una fabrica que produce productos de alta calidad, utilizando para esto la mejor materia prima, una confección y acabado que responda los más altos estándares de calidad , mostrando también productos exclusivos y diseños innovadores.

- **Organización**

La entidad en estudio está organizada por departamentos y secciones que se presenta en el siguiente organigrama.



**Figura 3.1.** Organigrama de la Fábrica "Intiwara"

**Fuente:** Elaboración Propia

### 3.2. Análisis del Sistema Actual

A continuación se presenta los diferentes procesos que se realizan en departamento de producción objeto de estudio del presente trabajo.

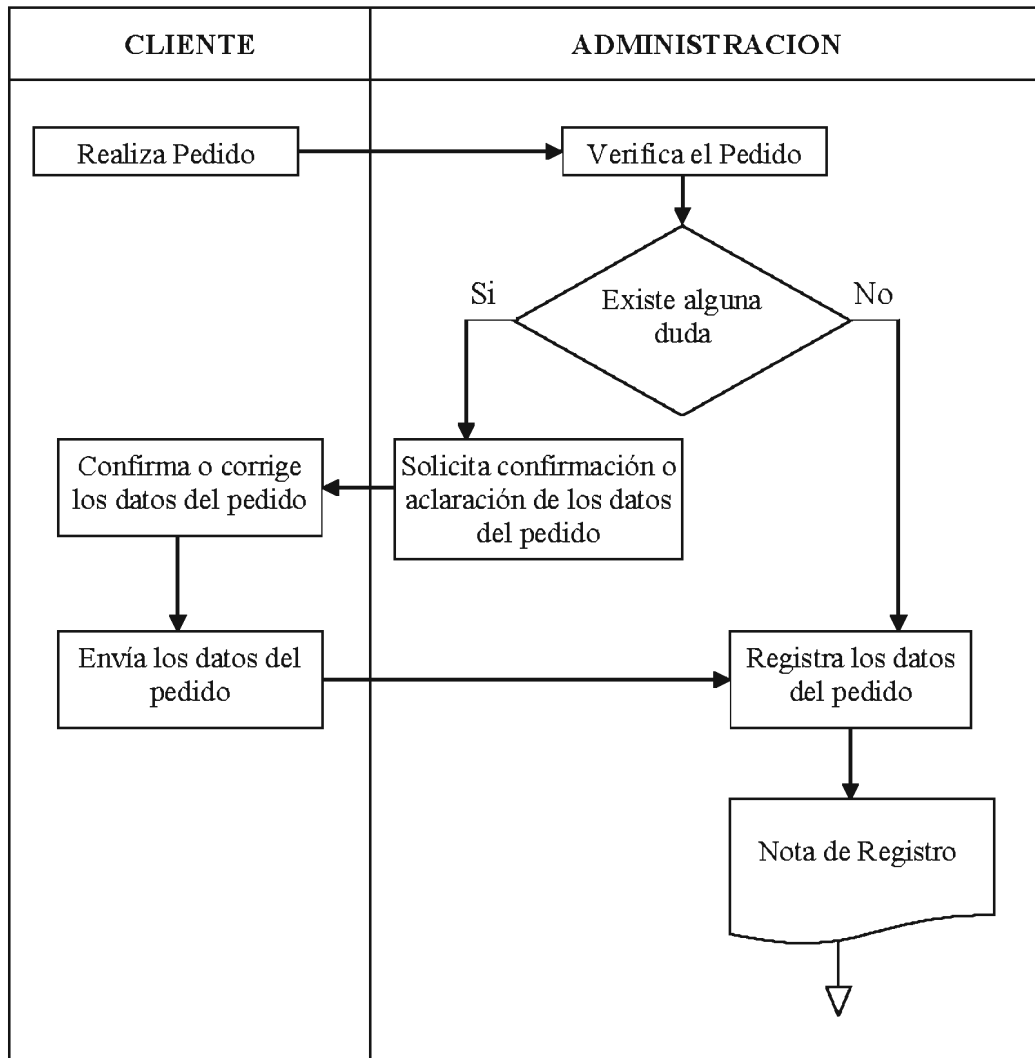


Tabla 3.1. Recepción de Pedidos

Fuente: Elaboración Propia

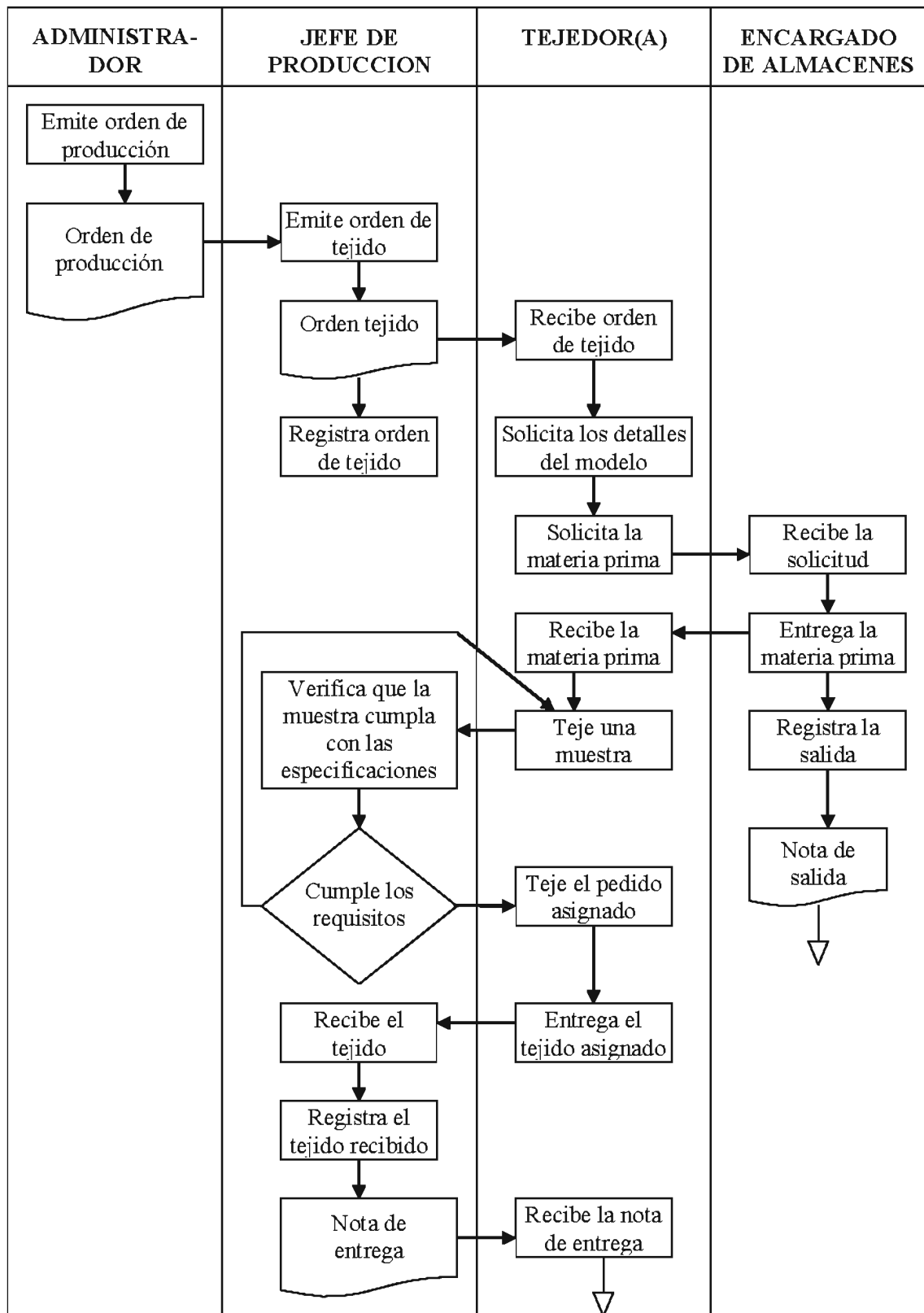


Tabla 3.2. Proceso Productivo – Tejido

Fuente: Elaboración Propia



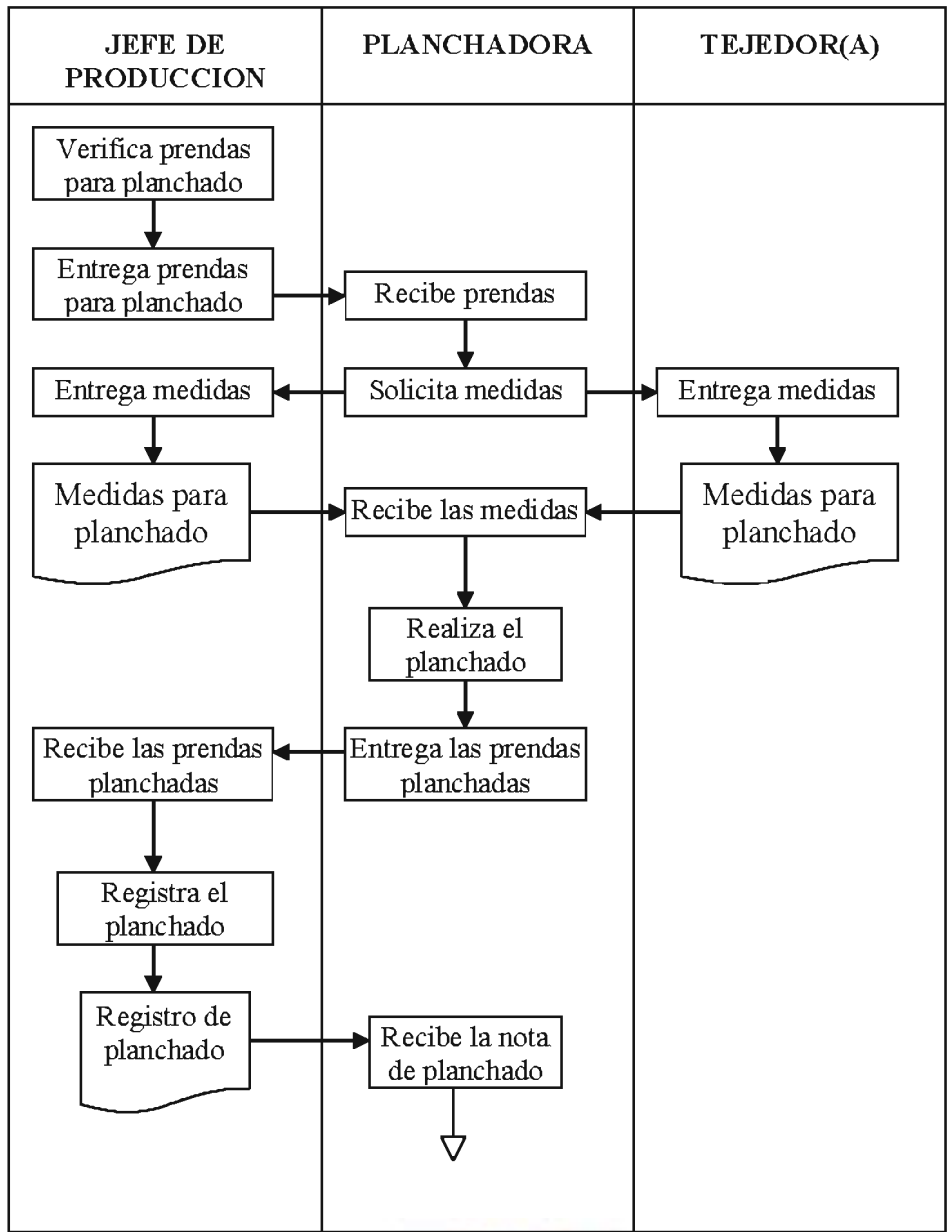


Tabla 3.4. Proceso Productivo – Planchado

Fuente: Elaboración Propia



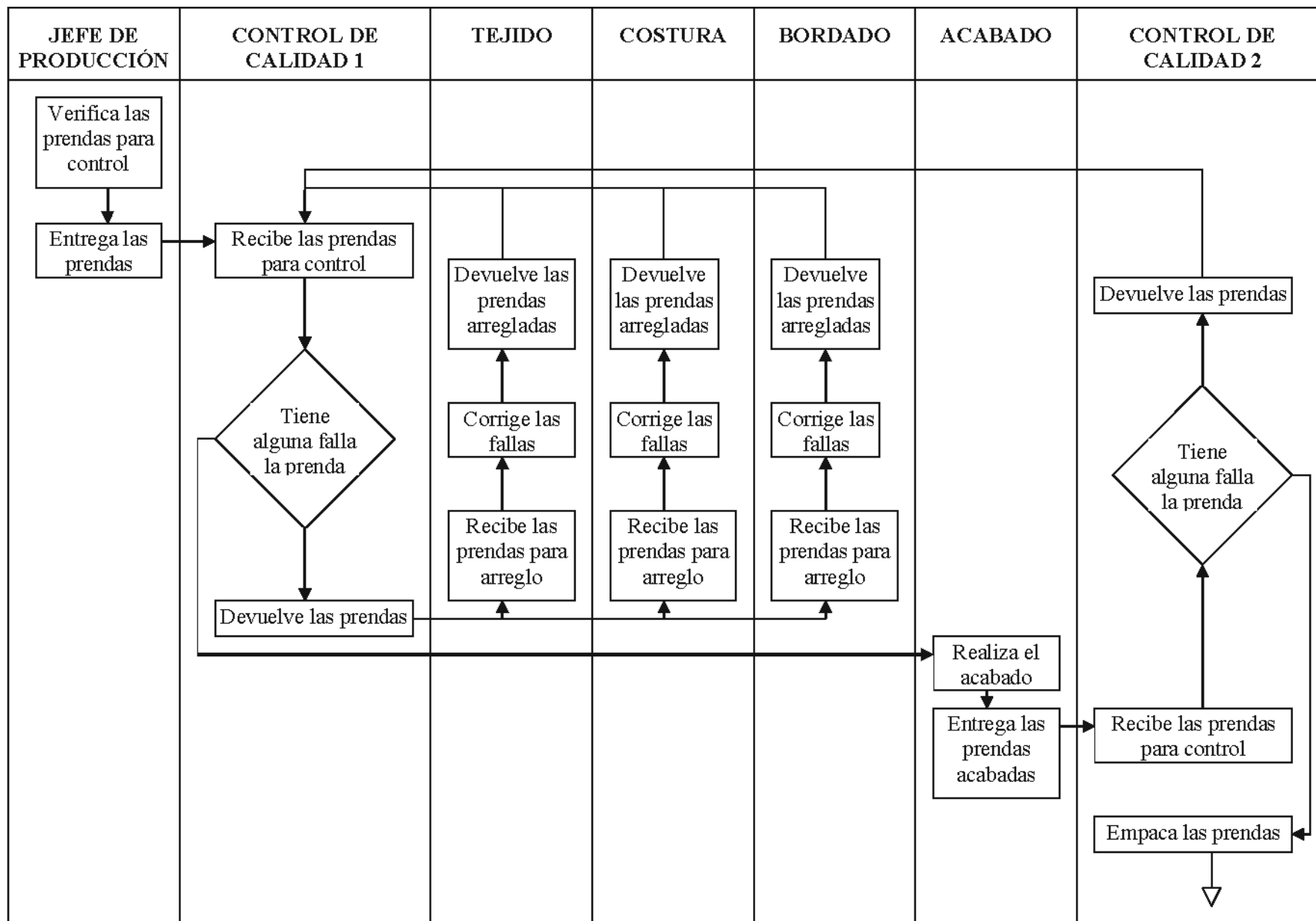


Tabla 3.5. Proceso Productivo – Control de Calidad y Acabado

Fuente: Elaboración Propia

Luego de haber examinado los diferentes procesos realizados en departamento de producción y sus diferentes secciones podemos mencionar los siguientes aspectos relacionados al manejo de la información:

- El registro de pedidos, se realiza de acuerdo al formato enviado por los clientes, no existiendo un formato o una norma general para dicho registro. Este registro se lo realiza en forma manual, o se conserva el fax, o se imprime el correo electrónico enviado por el cliente.
- Las ordenes de producción se lo realiza en forma manual, es registrada y clasificada por clientes.
- La asignación de trabajo (tejido, costura, planchado y bordado) se lo realiza en hojas sueltas que se entrega a cada empleado.
- La recepción del trabajo realizado (tejido, costura, planchado y bordado) se registra en el cuaderno personal de control de producción de los empleados.
- La entrega de materia prima a los tejedores es registrado en el cuaderno de producción de cada tejedor y en el cuaderno de salidas del encargado de almacenes.

Como se puede ver, el registro de los diferentes trabajos o transacciones en las diferentes secciones de producción se realiza en forma manual, lo que dificulta tener un control adecuado de este departamento.

Por otra parte la planificación de la producción se lo realiza basado en la experiencia adquirida en los años de trabajo de los administradores y propietarios de la fábrica.

### **3.3. Diseño del Nuevo Sistema**

El nuevo sistema a desarrollar se denomina “Sistema de Planificación y Control de la Producción de la Fábrica Intiwara”, (PCP – “Intiwara”).

#### **3.3.1. Especificación de Requerimientos**

Luego de la etapa de comunicación con el cliente (en este caso los administradores y propietarios de la fábrica), el análisis de tareas realizadas en el proceso productivo se pudo establecer los siguientes requerimientos del sistema:

Función	Gestión de Archivos
Nro.	Requerimientos
1	Automatizar el registro de los diferentes modelos que se producen en la fábrica, incluyendo el diseño, medidas, materia prima necesaria, combinaciones y las estimaciones de peso y tiempo de producción.
2	Automatizar la gestión de los archivos de los empleados (fijos y eventuales) y de usuarios
3	Automatizar el registro de pedidos utilizando un único formato.

**Tabla 3.6.** Especificación de Requerimientos – Gestión de Archivos

Fuente: Elaboración Propia

Función	Planificación de la Producción
Nro.	Requerimientos
1	Realizar el cálculo del tiempo requerido para poder cumplir con un determinado pedido
2	Realizar la planificación automática de la producción de un determinado pedido, teniendo en cuenta la disponibilidad de materiales y de empleados.
3	Generar diferentes alternativas de producción, tomando como variables las horas de trabajo y la cantidad de empleados.
4	Realizar el cálculo de materia prima necesaria para la producción de un determinado pedido.

**Tabla 3.7.** Especificación de Requerimientos – Planificación de la Producción

Fuente: Elaboración Propia

Función	Control de la Producción
Nro.	Requerimientos
1	Registrar la asignación de trabajo (tejido, costura, planchado, acabado y bordado), teniendo en cuenta la planificación realizada previamente.
2	Registrar la recepción de trabajo (tejido, costura, planchado, acabado y bordado), teniendo en cuenta la asignación realizada al empleado. Considerando también si cumple con la planificación realizada.
3	Realizar el control de la producción en forma individual y por secciones, tomando como parámetro la planificación realizada.

**Tabla 3.8.** Especificación de Requerimientos – Control de la Producción

Fuente: Elaboración Propia

<b>Función</b>	<b>Control de Almacenes</b>
<b>Nro.</b>	<b>Requerimientos</b>
<b>1</b>	Registrar la entrada y salida de materia prima.
<b>2</b>	Mantener un control de la existencia de materia prima generando alertas cuando la existencia de un determinado producto llegue a un determinado nivel.
<b>3</b>	Generar la orden de pedido de materia prima cuando la existencias no puedan cumplir la producción de los diferentes pedidos

**Tabla 3.9.** Especificación de Requerimientos – Control de Almacenes

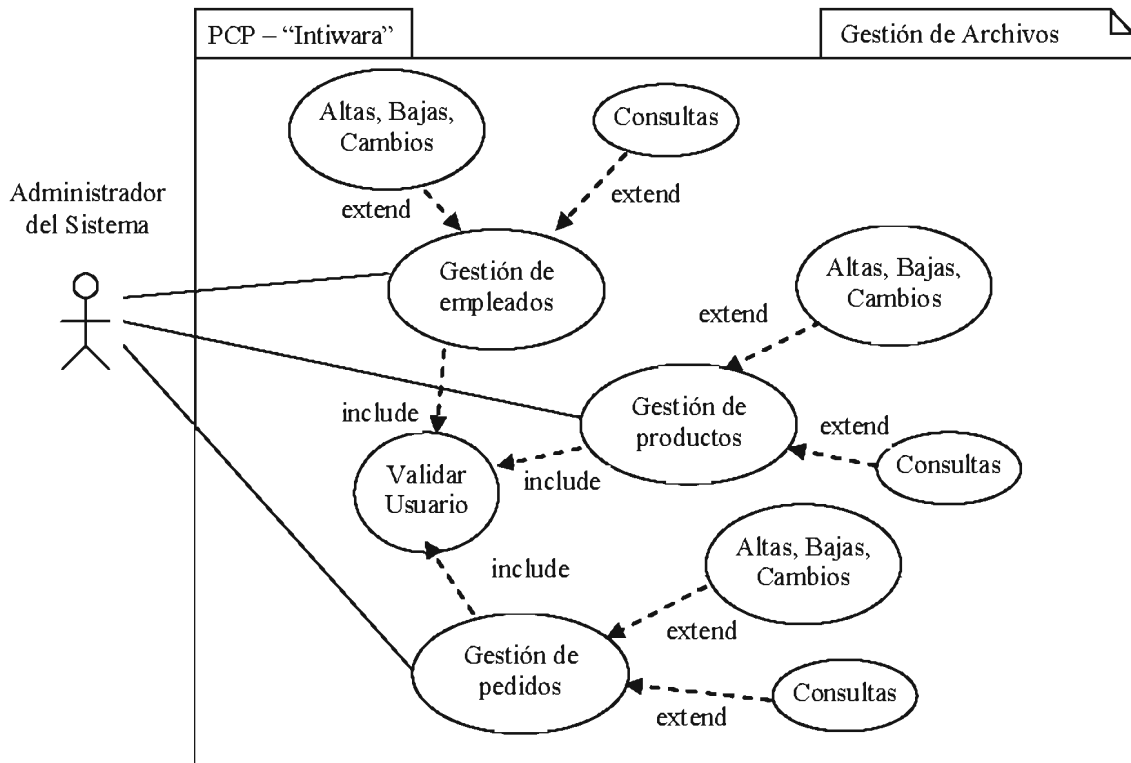
**Fuente:** Elaboración Propia

<b>Función</b>	<b>Reportes</b>
<b>Nro.</b>	<b>Requerimientos</b>
<b>1</b>	Generar reportes de la planificación realizada.
<b>2</b>	Generar reportes de la producción individual y por secciones, comparando con la planificación realizada.
<b>3</b>	Generar reportes de la existencia y el movimiento de materia prima.
<b>4</b>	Generar reportes del avance de la producción de un determinado pedido.
<b>5</b>	Generar reportes del material recibido por un determinado empleado.

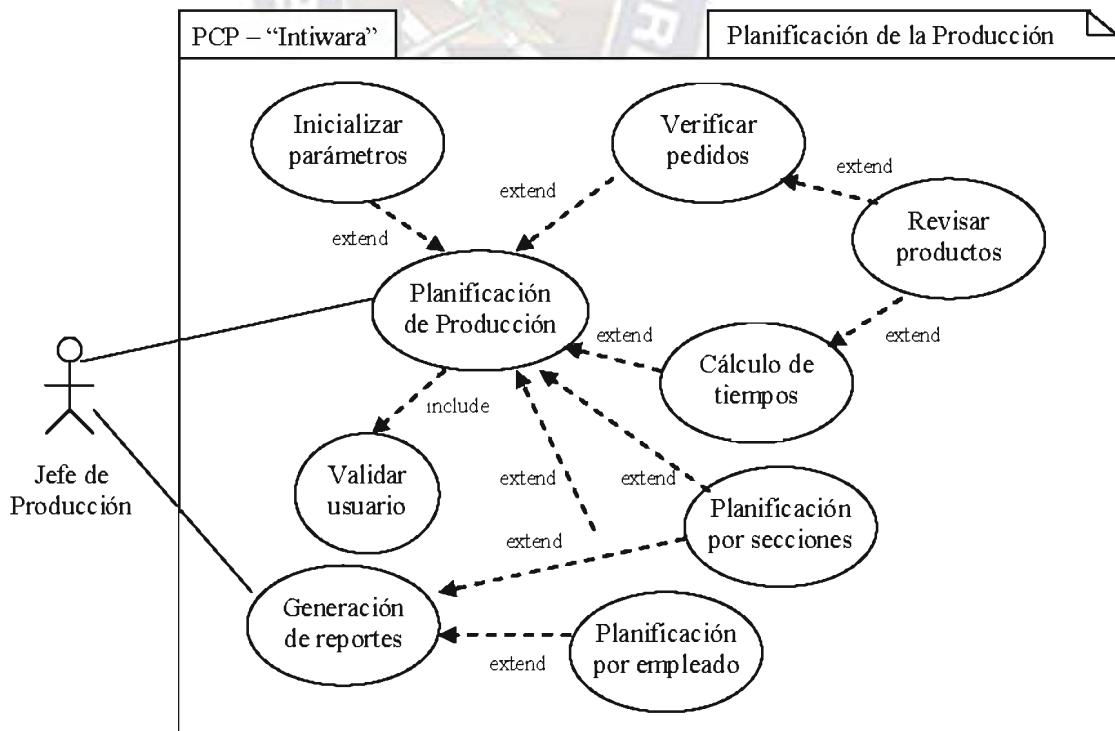
**Tabla 3.10.** Especificación de Requerimientos – Reportes

**Fuente:** Elaboración Propia

### **3.3.2. Casos de Uso**



**Figura 3.2.** Caso de Uso – Gestión de Archivos  
Fuente: Elaboración Propia



**Figura 3.3.** Caso de Uso – Planificación de la Producción  
Fuente: Elaboración Propia

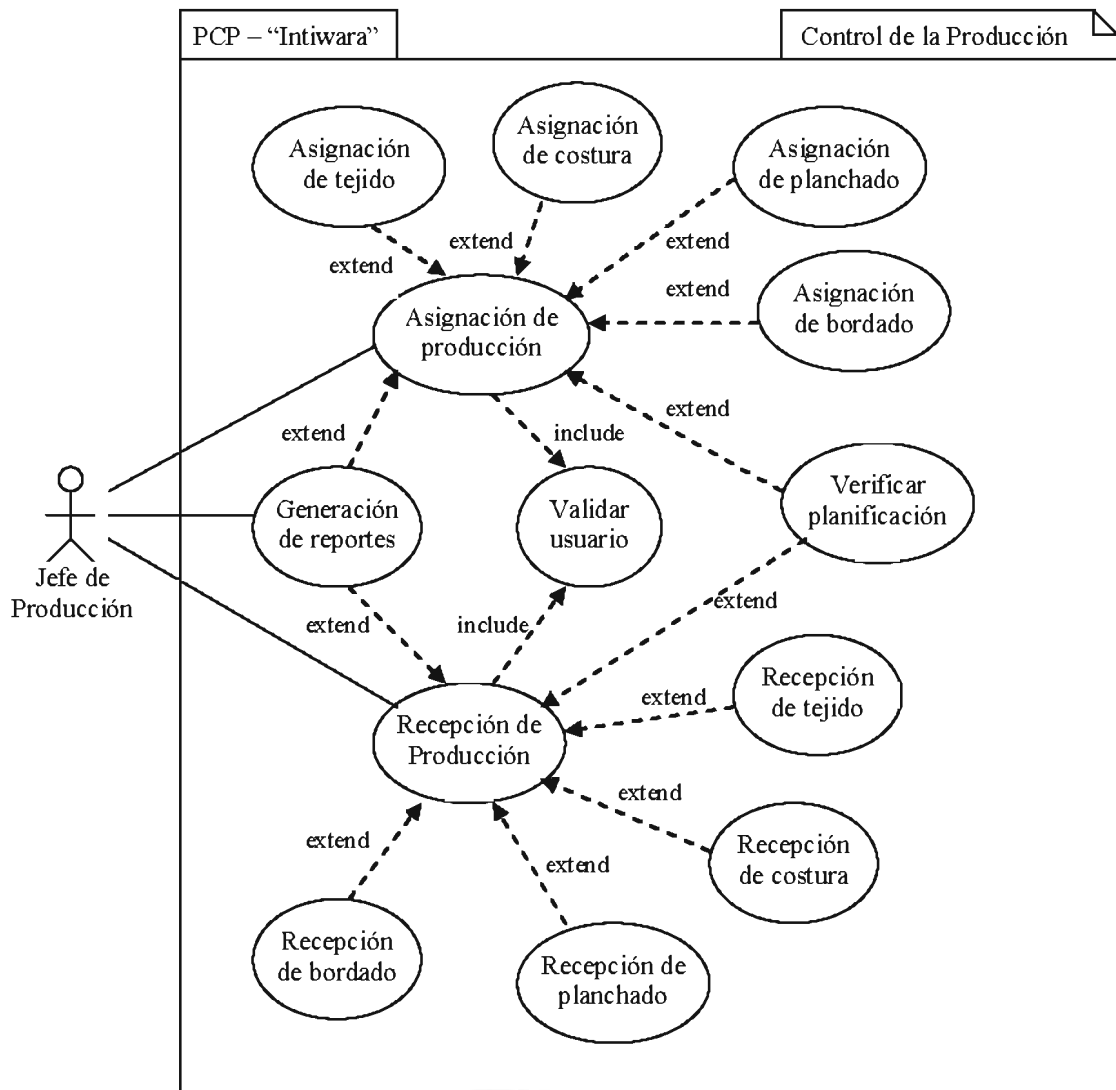


Figura 3.4. Caso de Uso – Control de la Producción

Fuente: Elaboración Propia

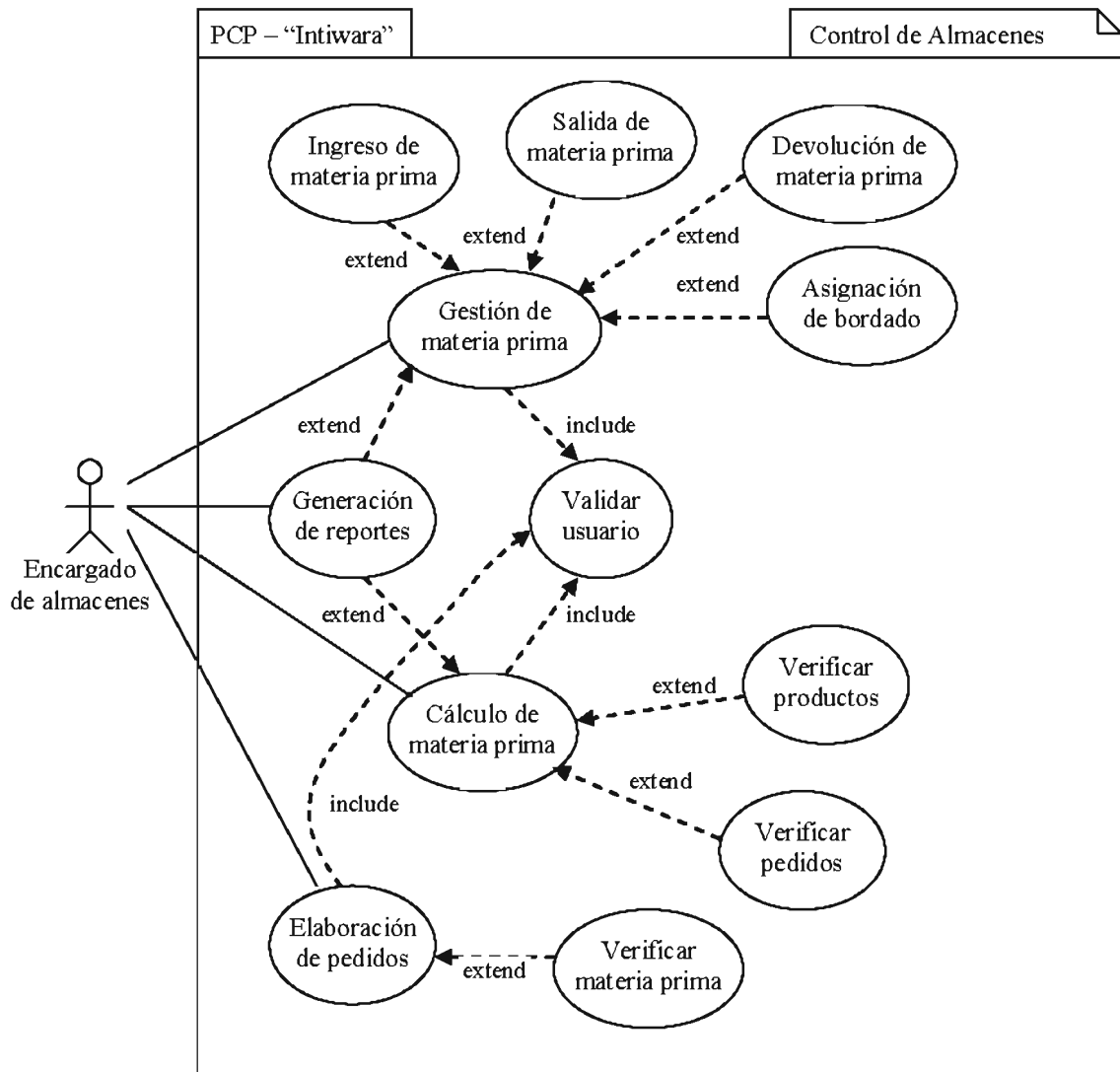


Figura 3.5. Caso de Uso – Control de Almacenes

Fuente: Elaboración Propia

### 3.3.3. Arquitectura del sistema

Una vez realizada el análisis del nuevo sistema se definió la arquitectura de tres capas, la cual se origina de la arquitectura cliente – servidor. La interfaz gráfica está conformada por diversas pantallas que conforman la interfaz del usuario, la interfaz lógica está desarrollada por código PHP que conecta las acciones del usuario con la base de datos que es la tercera capa, administrada en este caso por MySQL.

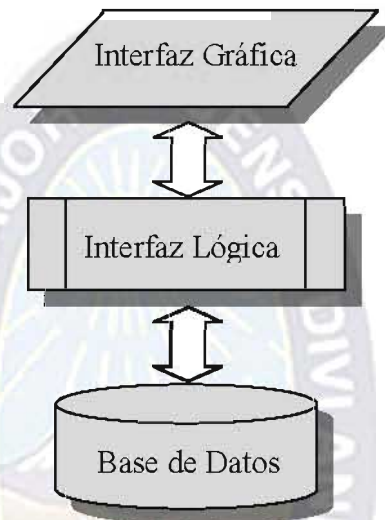


Figura 3.6. Arquitectura del Sistema

Fuente: Elaboración Propia

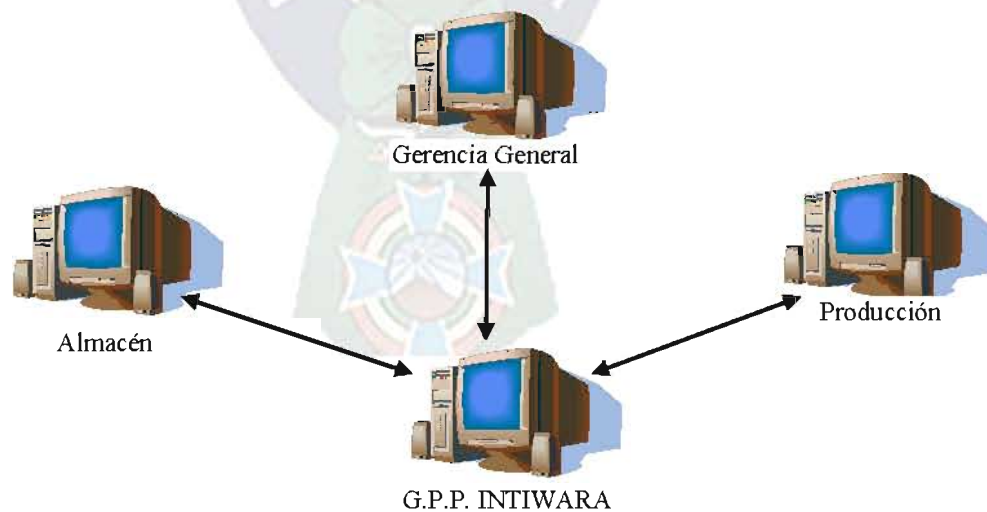


Figura 3.7. Organización del Sistema

Fuente: Elaboración Propia



### 3.3.4. Diagrama de Clases

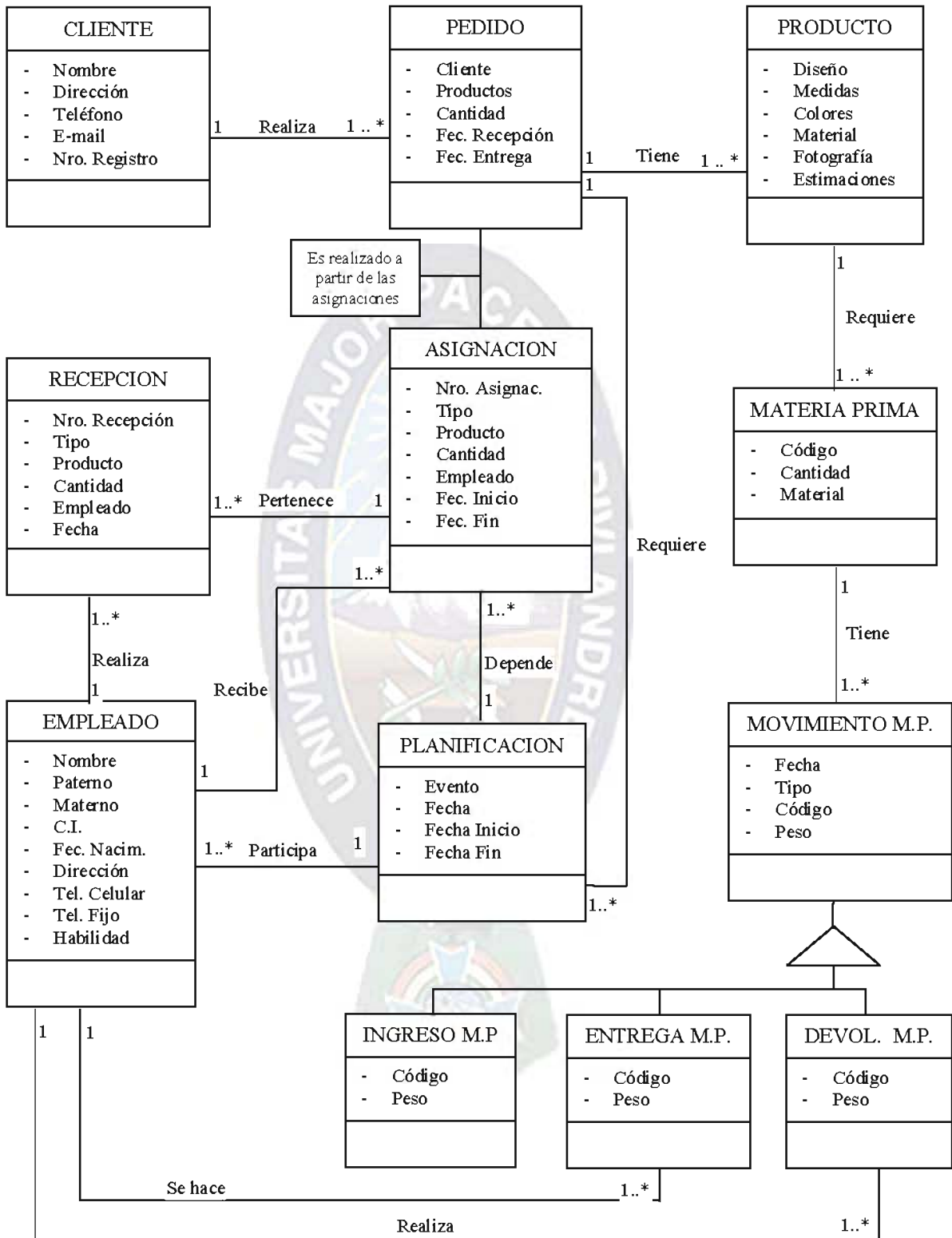
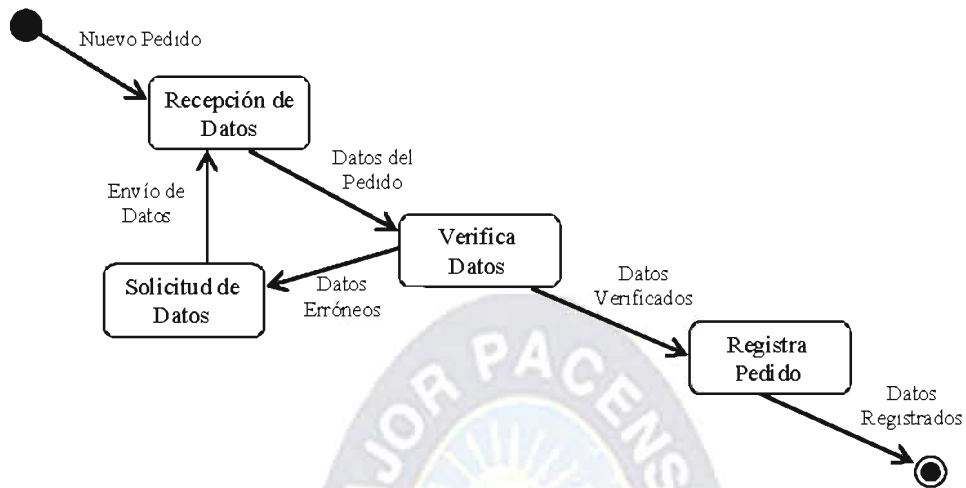


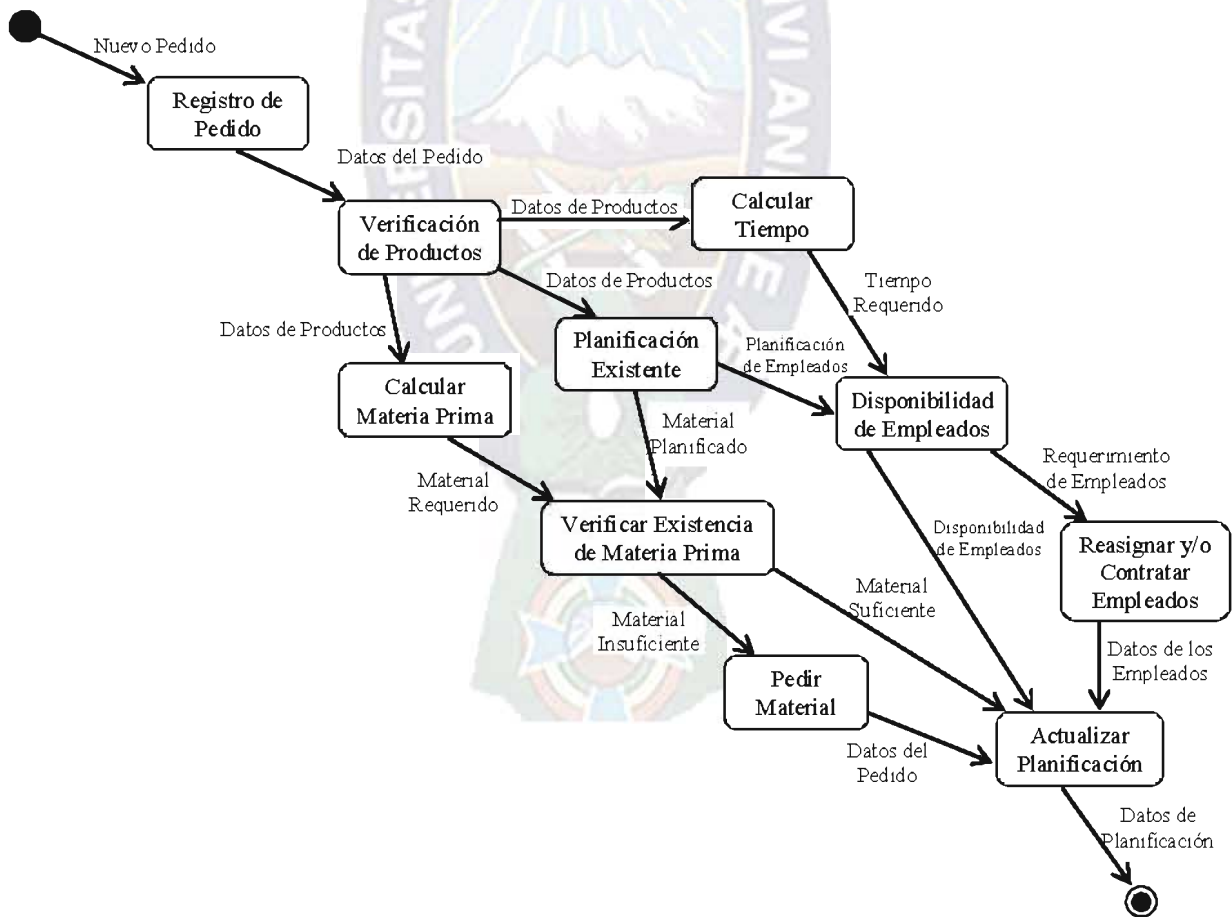
Figura 3.8. Diagrama de Clases

Fuente: Elaboración Propia

### 3.3.5. Diagrama de Transición de Estados



**Figura 3.9.** Diagrama Transición de Estados – Registro de Pedidos  
Fuente: Elaboración Propia



**Figura 3.10.** Diagrama Transición de Estados – Planificación  
Fuente: Elaboración Propia

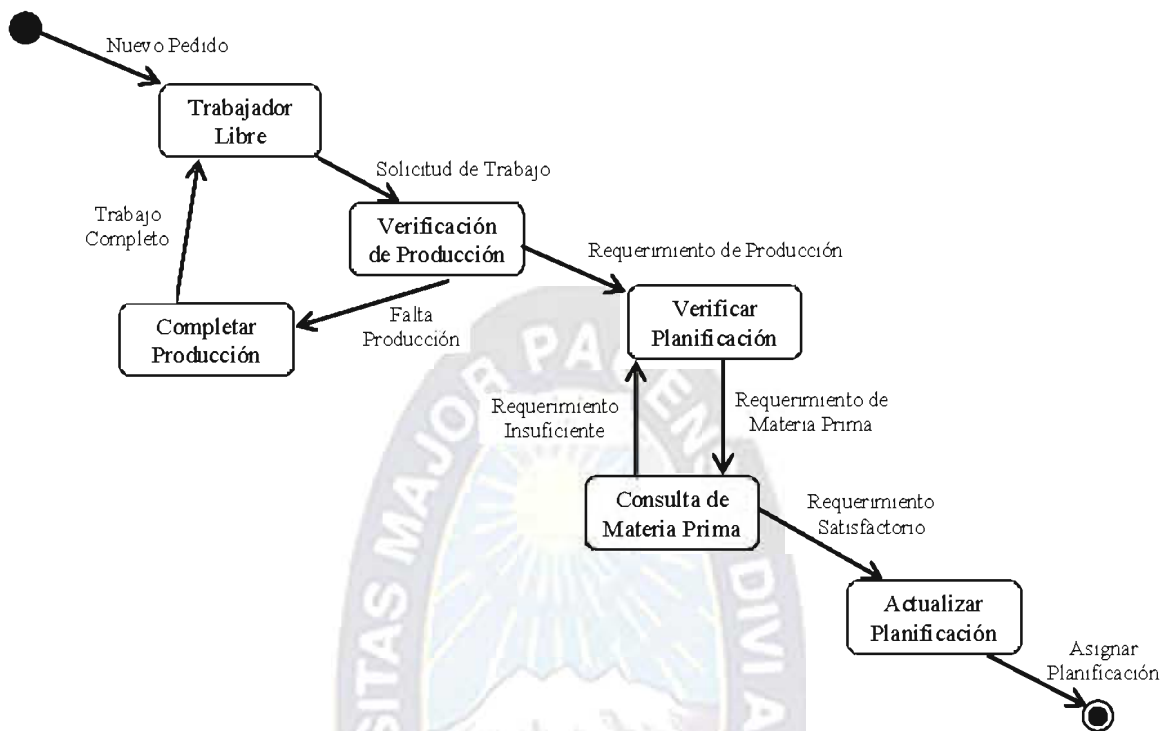


Figura 3.11. Diagrama Transición de Estados – Asignación de Trabajo

Fuente: Elaboración Propia

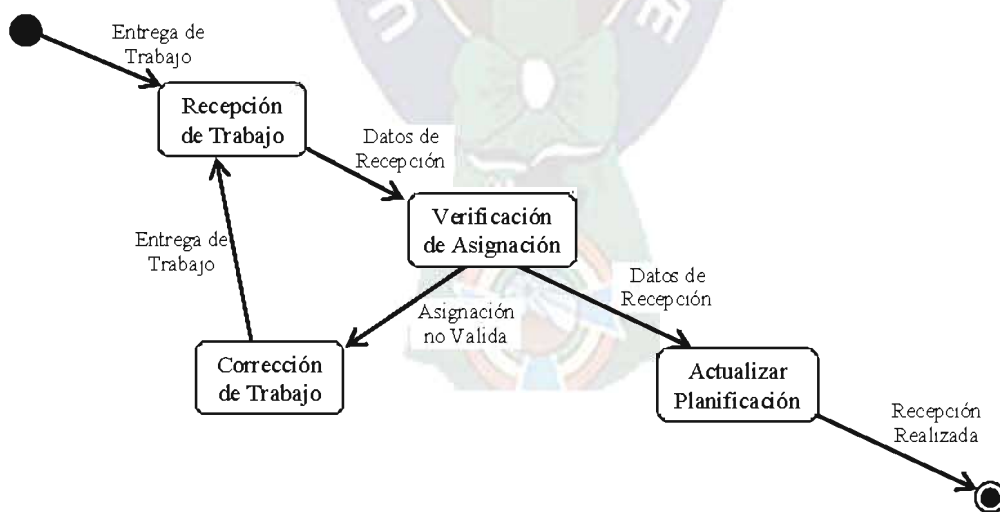
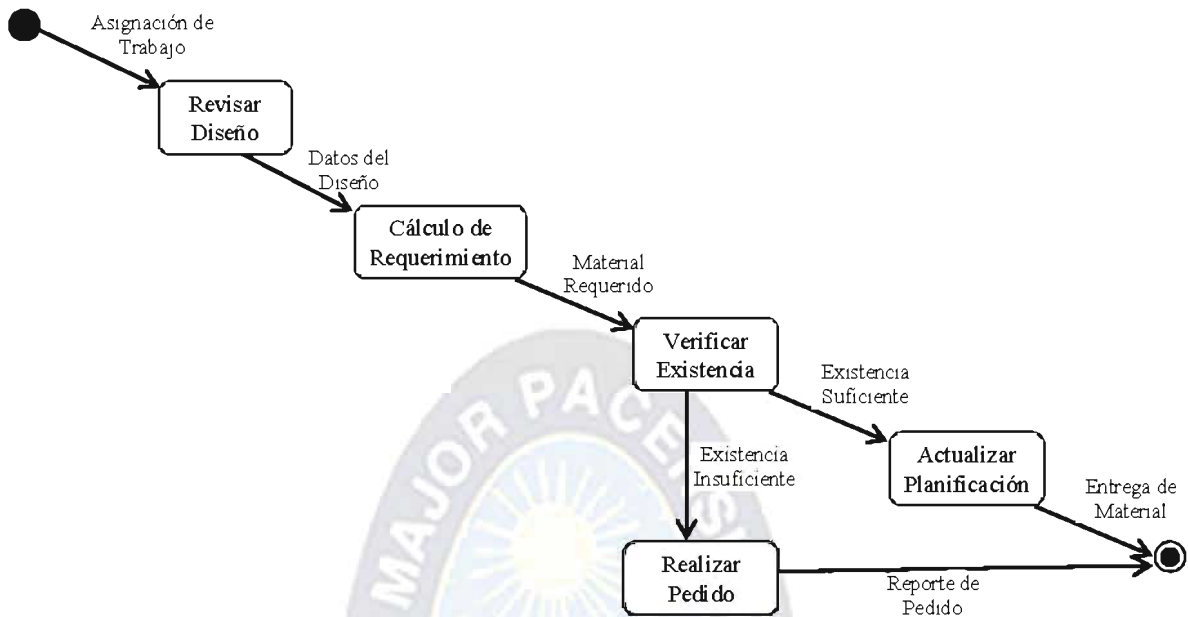


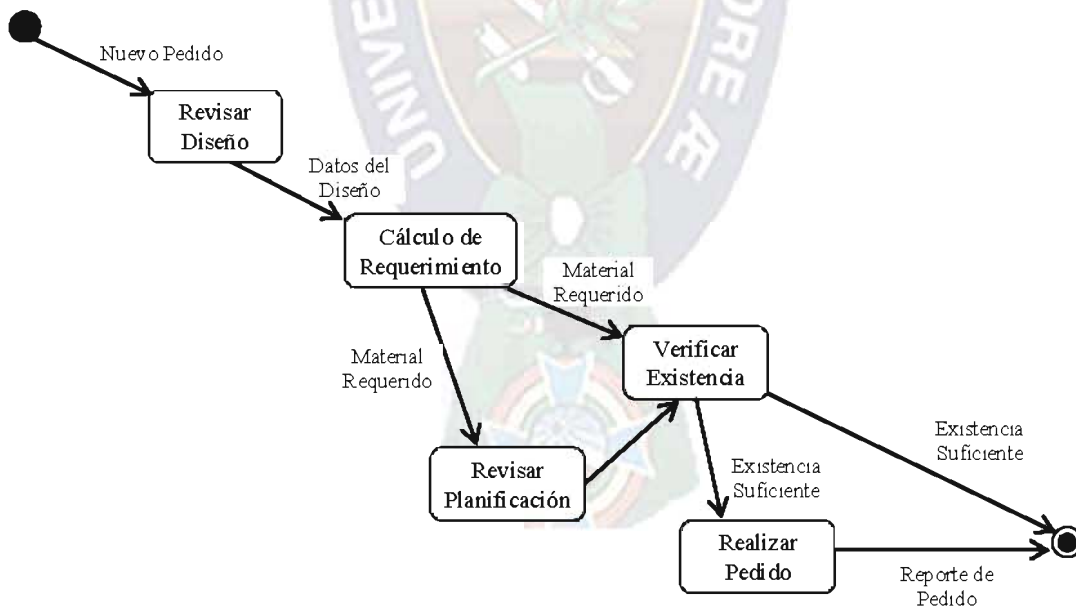
Figura 3.12. Diagrama Transición de Estados – Recepción de Trabajo

Fuente: Elaboración Propia



**Figura 3.13.** Diagrama Transición de Estados – Entrega de Materia Prima

**Fuente:** Elaboración Propia



**Figura 3.14.** Diagrama Transición de Estados – Pedido de Materia Prima

**Fuente:** Elaboración Propia

### 3.3.6. Diagrama de Secuencia

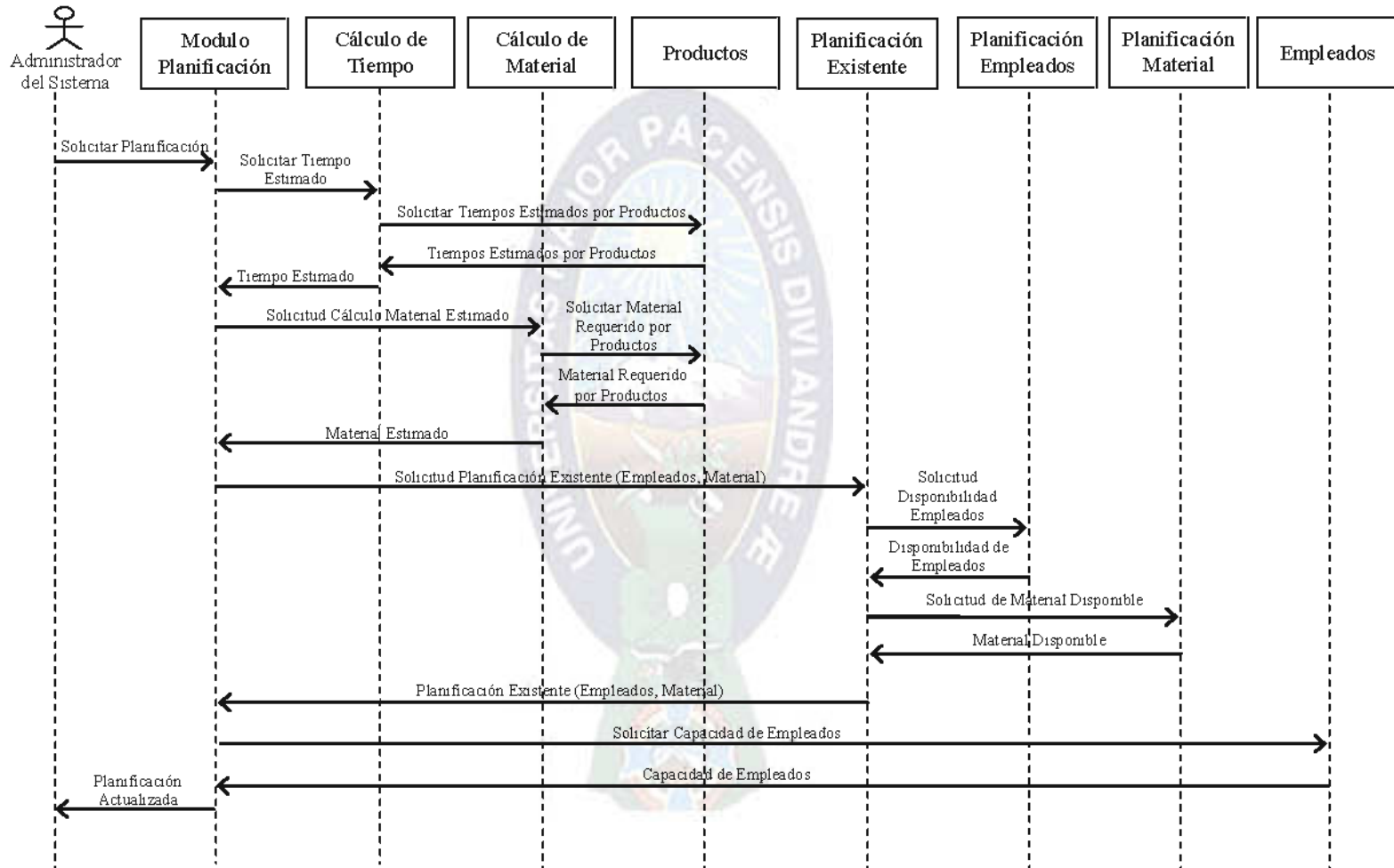


Figura 3.15. Diagrama de Secuencia – Planificación

Fuente: Elaboración Propia

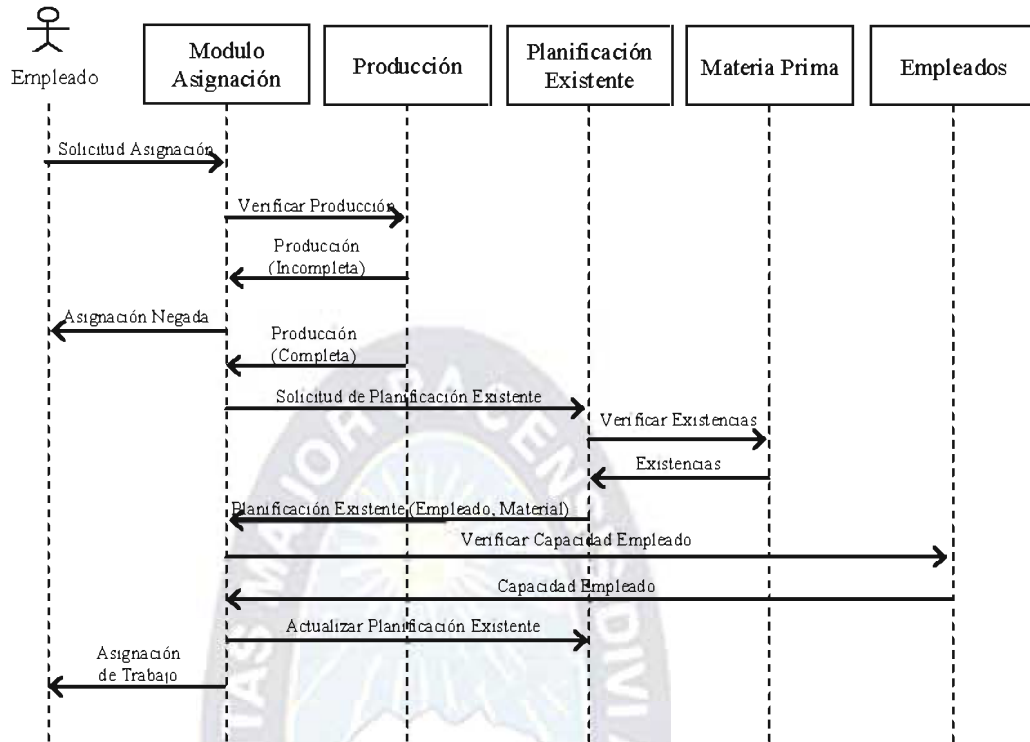


Figura 3.16. Diagrama de Secuencia – Asignación de Producción

Fuente: Elaboración Propia

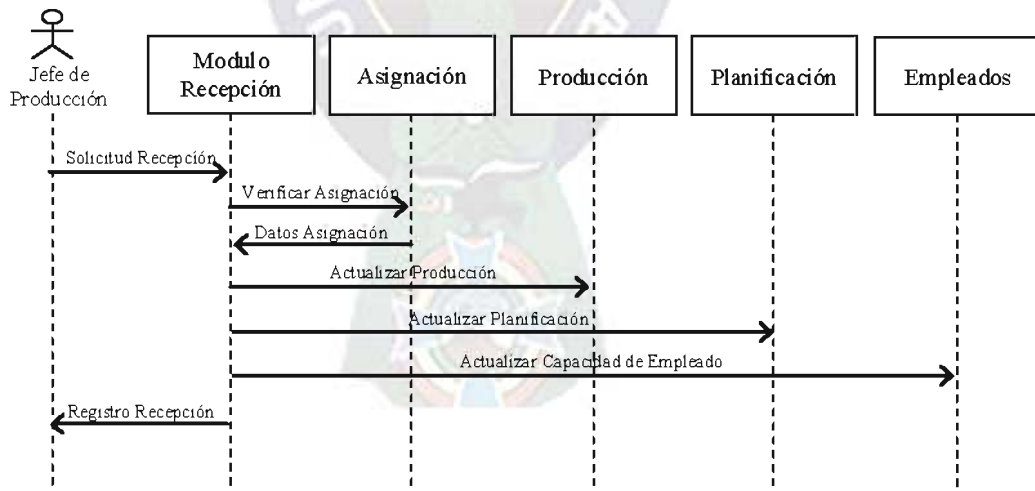


Figura 3.17. Diagrama de Secuencia – Recepción de Producción

Fuente: Elaboración Propia

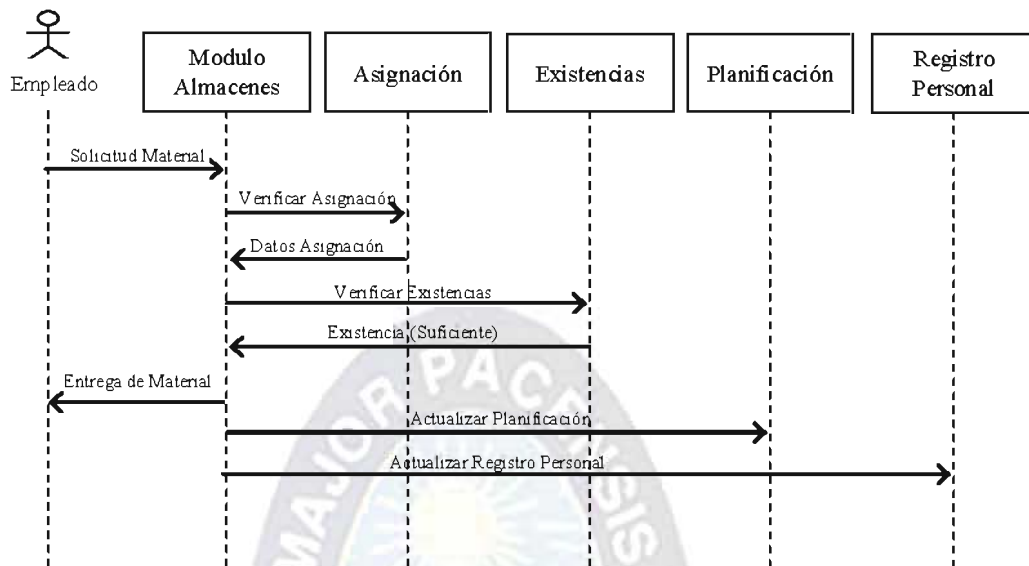


Figura 3.18. Diagrama de Secuencia – Entrega de Materia Prima

Fuente: Elaboración Propia

### 3.4. Diseño de Interfaz

A continuación se muestran las interfaces gráficas de usuario, para que este pueda interactuar con el sistema:

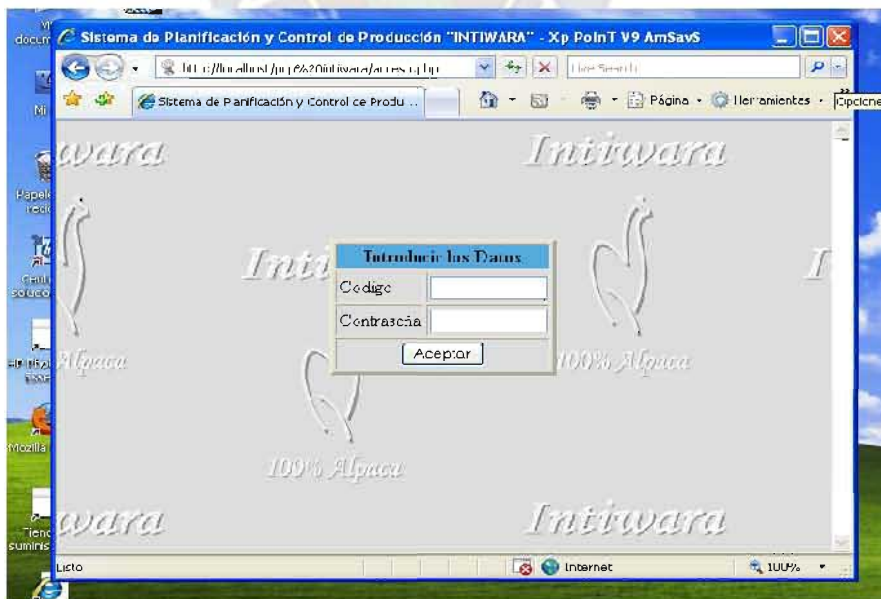


Figura 3.19. Diseño de Interfaz – Pantalla de Acceso

Fuente: Elaboración Propia

La pantalla de acceso permite que el usuario pueda acceder al sistema, previa verificación del código y contraseña validos, en caso de que se ingresen datos erróneos o que el usuario no esté registrado no podrá ingresar al sistema.

Luego de ingresar los datos válidos, se muestra la siguiente pantalla que contiene el menú principal.

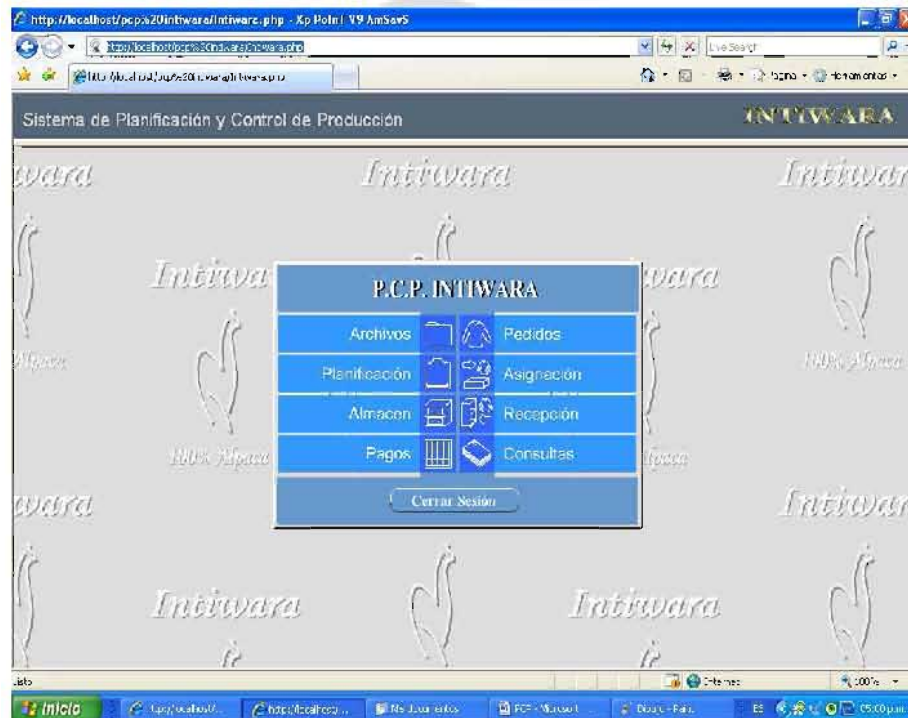


Figura 3.20. Diseño de Interfaz – Menú Principal

Fuente: Elaboración Propia

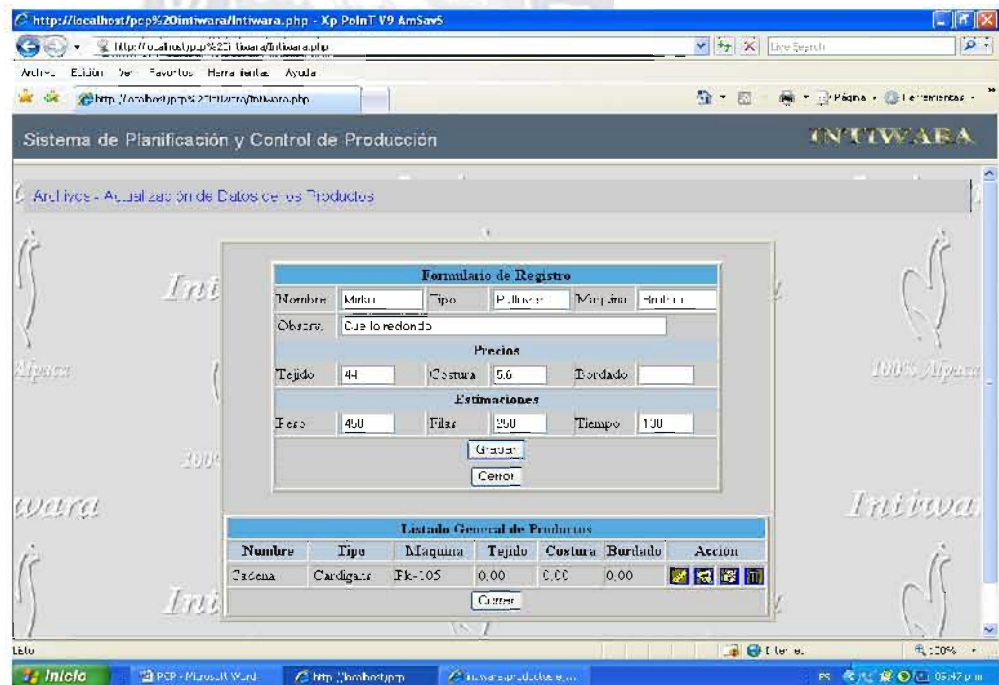
El menú principal contiene las siguientes opciones:

- **Archivos.** Esta opción permite realizar las altas, bajas y cambios de los archivos de Productos, Personal y Usuarios.
- **Pedidos.** Esta opción contiene las opciones de:
  - **Registrar,** un nuevo pedido.
  - **Actualizar,** los datos de los pedidos registrados.
  - **Asignar,** el pedido para su producción
- **Planificación.** Esta opción contiene las opciones de:



- **Producción**, permite realizar la planificación de la producción de un determinado pedido
- **Materia Prima**, permite planificar el uso o asignación de la materia prima en los diferentes pedidos.
- **Asignación**. Esta opción contiene las opciones de:
  - **Tejido**, permite el registro de la asignación de trabajo a un(a) determinado(a) tejedor(a).
  - **Costura**, permite el registro de la asignación del tejido realizado a un(a) determinado(a) costurero(a), para que realice la costura del producto.
  - **Planchado**, permite el registro de la asignación del tejido costurado a un(a) determinado(a) planchador(a), para que realice el planchado del producto.
  - **Bordado**, permite el registro de la asignación del producto a un(a) determinado(a) bordador(a), si así lo requiere el modelo.
  - **Acabado**, permite el registro de la asignación del producto a un(a) determinado(a) empleado(a), para el acabado final.
- **Recepción**. Esta opción contiene las opciones de:
  - **Tejido**, permite el registro de la recepción del tejido realizado por un(a) determinado(a) tejedor(a)
  - **Costura**, permite el registro de la recepción de los productos costurados realizada por un(a) determinado(a) costurero(a).
  - **Planchado**, permite el registro de la recepción del producto planchado por un(a) determinado(a) planchador(a).
  - **Bordado**, permite el registro de la recepción del producto bordado por un(a) determinado(a) bordador(a).
  - **Acabado**, permite el registro de la recepción del producto acabado por un(a) determinado(a) empleado(a).
- **Pagos**. Esta opción contiene las opciones de:
  - **Anticipos**, permite el registro del anticipo que se otorga a un determinado empleado.
  - **Producción**, permite el registro del pago por producción, que se realiza a un determinado empleado.

- **Mensual**, permite el registro del pago mensual realizado a un determinado empleado.
- **Consultas.** Esta opción contiene las opciones de:
  - **Archivos**, permite visualizar el contenido de los archivos de Productos, Personal y Usuarios
  - **Pedidos**, permite visualizar el estado y/o avance en que se encuentra un determinado pedido.
  - **Planificación**, permite visualizar la planificación realizada y el estado y/o avance en que se encuentra un determinado pedido.
  - **Producción**, permite visualizar la producción realizada de un determinado pedido, modelo, empleado o sección.
  - **Almacén**, permite visualizar el ingreso, entrega, devolución de materia prima, también se puede visualizar el movimiento de la materia prima desde el inventario inicial.



**Figura 3.21.** Diseño de Interfaz – Registro de Productos

Fuente: Elaboración Propia

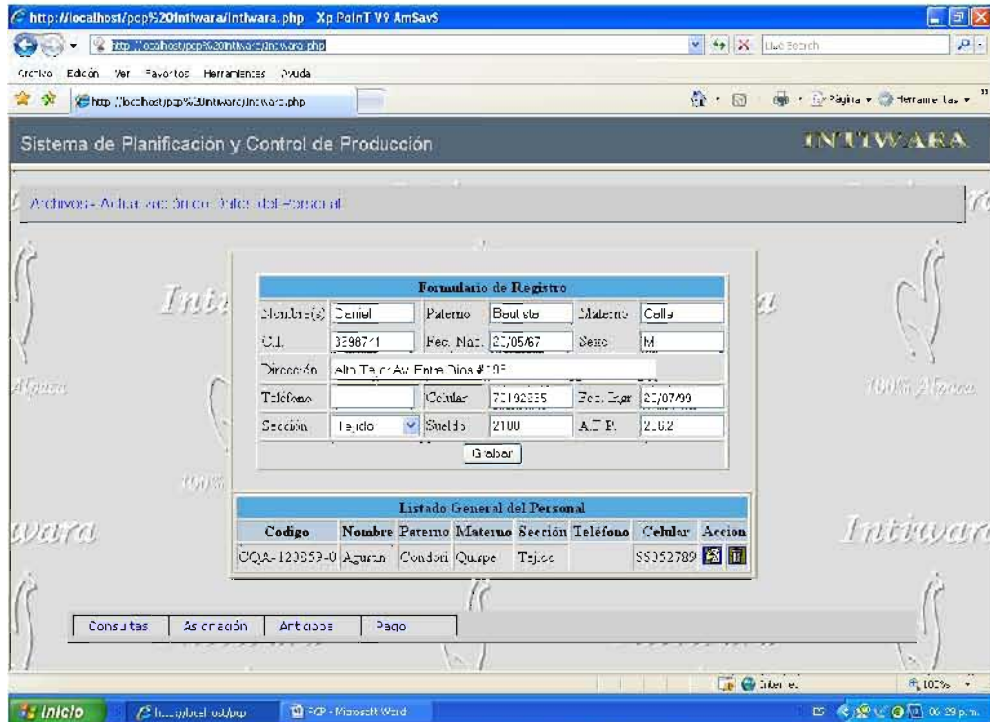


Figura 3.22. Diseño de Interfaz – Registro de Personal

Fuente: Elaboración Propia

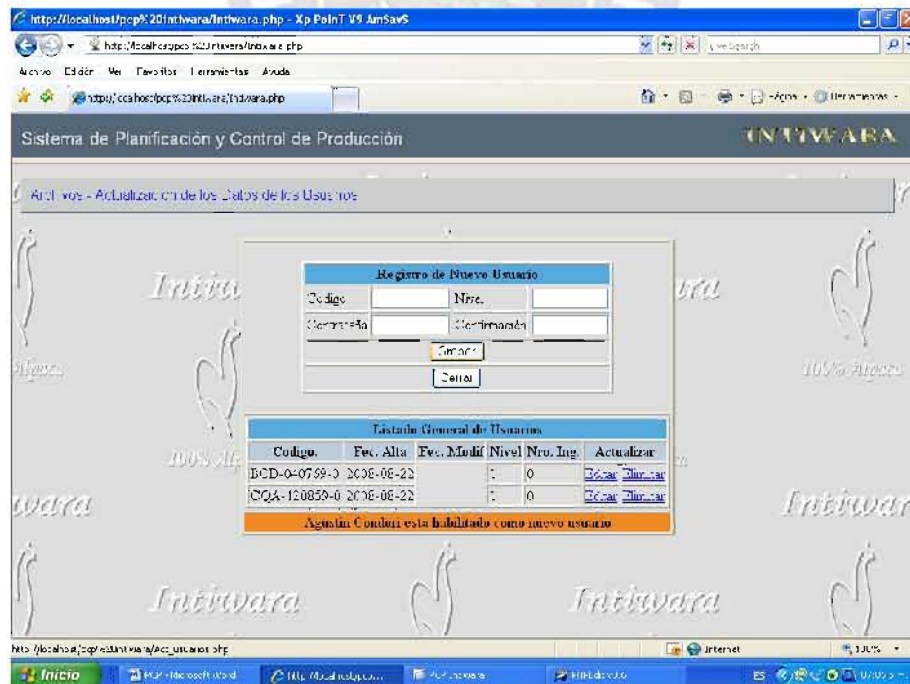


Figura 3.23. Diseño de Interfaz – Registro de Usuarios

Fuente: Elaboración Propia

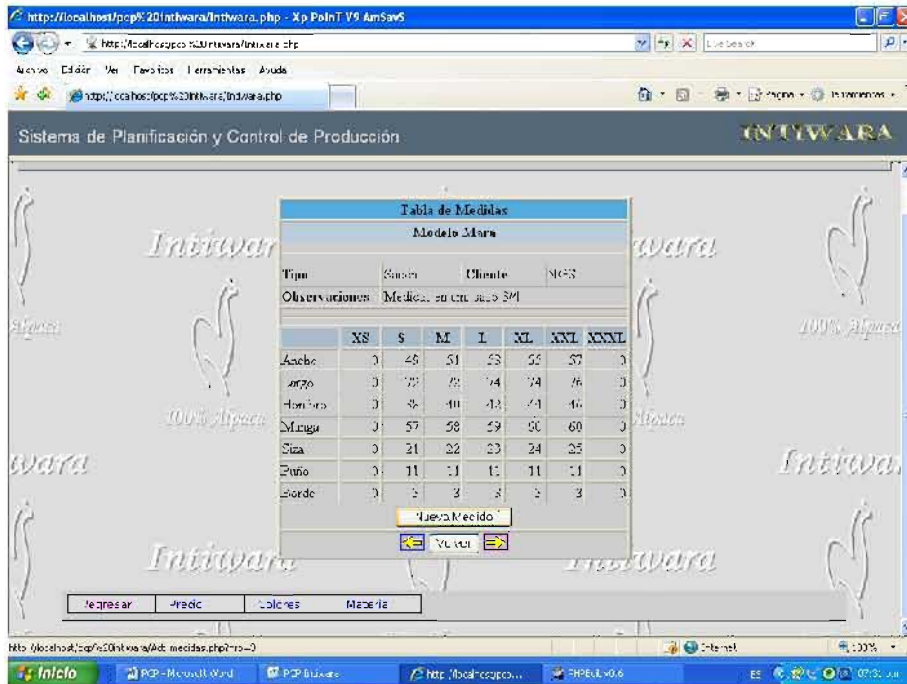


Figura 3.24. Diseño de Interfaz – Tabla de Medidas

Fuente: Elaboración Propia

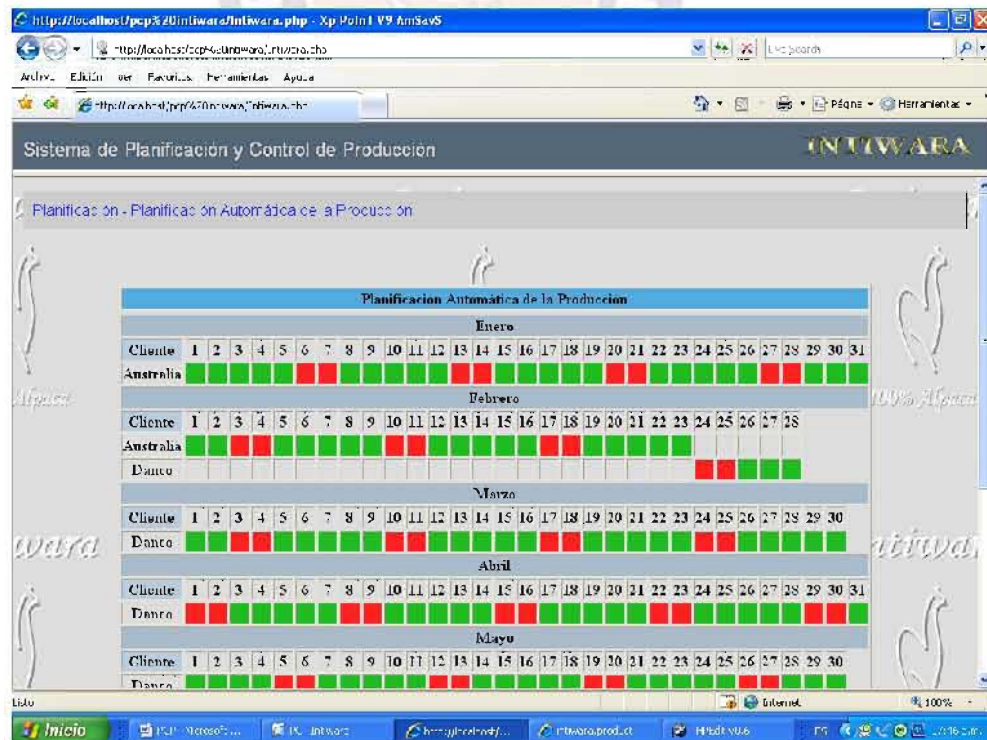


Figura 3.25. Diseño de Interfaz – Planificación de la Producción

Fuente: Elaboración Propia

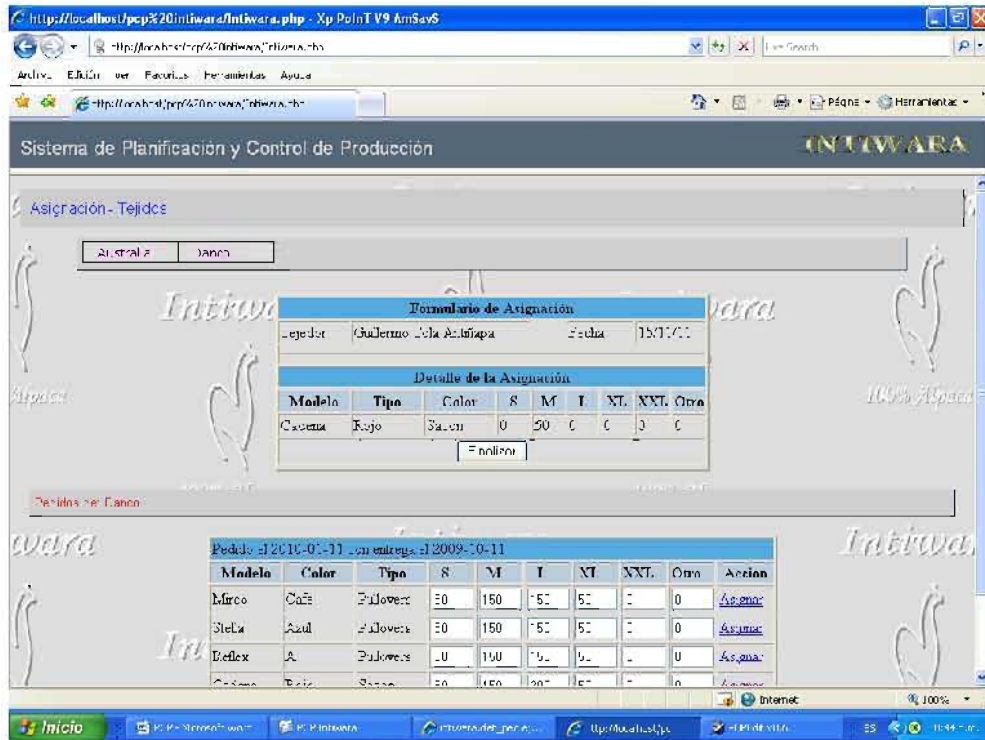


Figura 3.26. Diseño de Interfaz – Asignación de Tejido

Fuente: Elaboración Propia

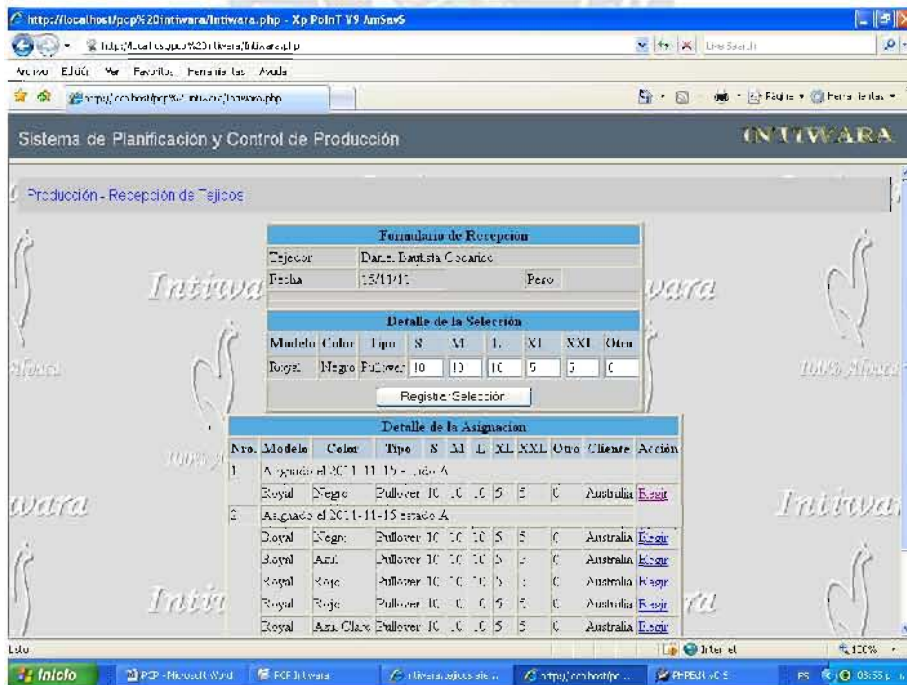


Figura 3.27. Diseño de Interfaz – Recepción de Tejido

Fuente: Elaboración Propia

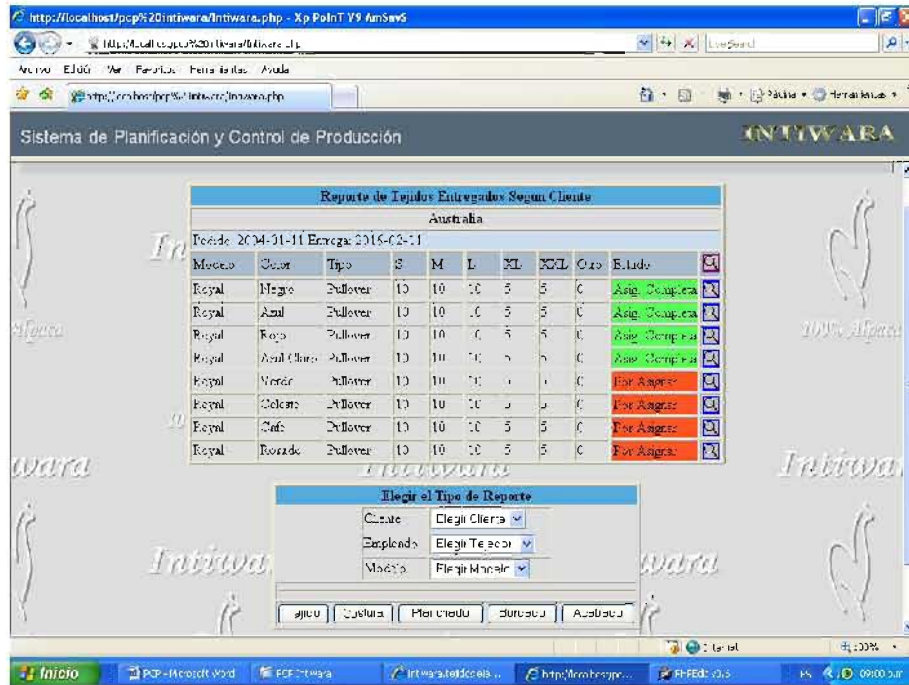


Figura 3.28. Diseño de Interfaz – Reporte de Producción

Fuente: Elaboración Propia

### 3.5. Requerimientos para la Implementación

Para la implementación del presente proyecto se requiere los siguientes componentes con las características mínimas descritas:

#### Hardware:

- Sistema Operativo:** Windows 98 o superior.
- Lenguaje de Programación:** PHP, HTML.
- Gestor de Base de Datos:** MySQL.

#### Software:

- Microprocesador:** Pentium III o superior
- Memoria:** 256 Mb.
- Espacio en Disco Duro** 80 Mb.
- Monitor** 14"
- Teclado, Mouse, Impresora**

#### 4. CALIDAD Y SEGURIDAD



## **4. CALIDAD Y SEGURIDAD**

### **4.1. Calidad del Sistema**

Los factores y atributos de la calidad presentados en el segundo capítulo se centran en el software como un todo y pueden utilizarse como indicación general de la calidad de una aplicación. Para ver el grado que se satisface un factor se propuso preguntas asociadas a cada factor o atributo.

Por otro lado los criterios de calidad tienen un carácter subjetivo, es decir, la métrica asignada al criterio como respuesta a las preguntas asociadas a los atributos de calidad pueden tomar valores como: Muy Bueno, Bueno, Regular, Malo o Muy Malo.

Luego de haber realizado la evaluación del sistema se pudieron obtener los siguientes resultados.

#### **4.1.1. Funcionalidad**

Este aspecto mide si el software satisface las necesidades planteadas, tomando en cuenta los atributos relacionados con este factor de la calidad.



Funcionalidad		
Factores	Grado Estimado	Criterios de Estimación
Adaptabilidad	Buena	* Se puede adaptar a los cambios con facilidad - Los módulos son relativamente independientes
Exactitud	Muy Buena	* Los resultados que se presentan son los esperados * Los diferentes módulos realizan tareas específicas
Cumplimiento	Buena	* Cubre con los requerimientos de los usuarios

**Tabla 4.1.** Funcionalidad del Sistema

**Fuente:** Elaboración Propia

#### 4.1.2. Usabilidad

Este aspecto mide la facilidad de uso por parte de los usuarios, teniendo en cuenta los aspectos relacionados con este atributo.

USABILIDAD			
Preguntas Realizadas	Usuario 1	Usuario 2	Usuario 3
¿La entrada de datos a través del teclado y del ratón es... ?	Buena	Muy Buena	Buena
¿La comprensión de las salidas es... ?	Buena	Muy Buena	Buena
¿La localización de las operaciones son... ?	Regular	Buena	Buena
¿Las metáforas presentadas en la interfaz es... ?	Muy Buena	Muy Buena	Muy Buena
¿La estética de las paginas son... ?	Muy Buena	Muy Buena	Buena

**Tabla 4.2.** Usabilidad del Sistema

**Fuente:** Elaboración Propia

Por las respuestas obtenidas podemos concluir que la usabilidad del sistema es BUENA

#### 4.1.3. Eficiencia

Los criterios de eficiencia tienen un carácter subjetivo, es decir, el nivel de eficiencia en cuanto a los recursos (hardware y software) usados por parte del sistema tiene los siguientes valores.

- **Alta.**  
Significa que el sistema consume recursos informáticos de manera moderada, entendiéndose, que es sistema permite la ejecución de otros programas de manera simultánea sin provocar la sobrecarga del ordenador.
- **Media**  
Significa que el sistema acapara algunos recursos informáticos sobrecargando el ordenador en caso de ejecutar otra aplicación.
- **Baja.**  
Significa que el sistema acapara varios recursos informáticos de forma exclusiva, sobrecargando y provocando errores en el ordenador.

USO DE RECURSOS			
Recursos	GPP Intiwara	AppServ	Total
Memoria RAM	33,9 Kb.	38,1 Kb.	72 Kb.
Almacenamiento en Disco Duro	600 Kb.	30,78 Mb.	31.36 Mb.

**Tabla 4.3.** Uso de Recursos

Fuente: Elaboración Propia

Por lo observado en la Tabla 4.4. el uso de recursos por parte del software es mínimo y por tanto es posible la ejecución de otros programas sin ocasionar ningún conflicto entre los mismos. Por lo tanto la eficiencia del sistema es ALTA.

#### 4.1.4. Facilidad de Recibir Mantenimiento

La estructura y diseño del sistema permite realizar el siguiente análisis:

Facilidad de Mantenimiento		
Factores	Grado Estimado	Criterios de Estimación
Estabilidad	Buena	* Cubre todos los requerimientos de los usuarios
Facilidad de Análisis	Muy Buena	* Los diferentes módulos son relativamente pequeños * El código cuenta con comentarios que facilita su análisis
Facilidad de Cambios	Buena	* Los diferentes módulos realizan una tarea específica
Facilidad de Pruebas	Buena	* Los módulos pueden ejecutarse independientemente

**Tabla 4.4.** Facilidad de Mantenimiento

**Fuente:** Elaboración Propia

Por lo descrito en la Tabla 4.5. podemos concluir que la facilidad de recibir mantenimiento es BUENA.

#### **4.1.5. Portabilidad**

Este criterio de calidad tiene un carácter técnico, que depende de la siguiente fórmula:

$$P(X, x) = \frac{x}{X}; \quad 0 \leq P \leq 1$$

Donde:

P: Es el valor asociado a la portabilidad del software desarrollado que puede tomar valores de 0 a 1, siendo 0 el valor mínimo 1 el valor más alto de portabilidad.

X: Es el número de ordenadores disponibles en la empresa

x: Es el número de ordenadores donde el software se ejecuta correctamente en la institución.

Calculando tenemos:

$$X = 4; x = 4 \quad \Rightarrow \quad P = \frac{4}{4} = 1$$

Como el resultado podemos concluir que el software tiene una portabilidad aceptable dentro la empresa.

Por lo descrito anteriormente, tomando en cuenta los diferentes criterios de calidad, podemos concluir que se ha desarrollado un software una calidad BUENA.

## **4.2. Seguridad del Sistema**

La seguridad del sistema es una de las aplicaciones más requeridas por los usuarios por lo que se cuenta con:

### **4.2.1. Seguridad de Acceso.**

Para la seguridad de acceso, el sistema cuenta con código implementado en cada una de las páginas que no permite el ingreso a las mismas sin antes haber sido autenticado el usuario, y este es enviado a la página de acceso al sistema.

Cabe hacer notar también que los equipos se encuentran en secciones restringidas a la mayoría de los empleados y que solo pueden acceder a las mismas pocas personas autorizadas.

#### **4.2.2. Autenticación de Usuarios.**

El sistema valida cada acceso al sistema comparando los datos ingresados por el usuario (Código y Contraseña), con la base de datos en la que se encuentran registrados los usuarios que pueden acceder al sistema.

Luego de haber sido autenticado el usuario el sistema registra todas las acciones realizadas por el usuario.





## 5. CONCLUSIONES Y RECOMENDACIONES

## 5. CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones.

El sistema para la Gestión del Proceso Productivo de la Fábrica “INTIWARA”, luego de su implementación en el departamento de producción cumplió satisfactoriamente el objetivo general planteado en el capítulo 1.

“Desarrollar e implementar software para facilitar la planificación y el control de la producción en la fábrica Intiwara”.

Para conseguir este objetivo general también se cumplió con los objetivos específicos planteados para el presente proyecto.

- Gestión de los diferentes archivos.  
El sistema proporciona la posibilidad de actualizar (adicionar, eliminar y modificar los registros) los diferentes archivos involucrados en el proceso productivo.
- Planificación de la producción.

El sistema proporciona el módulo de planificación tomando en cuenta para esta actividad el tiempo requerido para la producción de los diferentes modelos, la destreza y disponibilidad de los empleados.

- **Gestión de almacenes.**  
El módulo de almacenes permite el registro de un inventario inicial y del movimiento de la materia prima, es decir, registra el ingreso (compra y recepción), la entrega (a un determinado empleado) y la devolución (realizada por un empleado) de la materia prima.
- **Asignación y recepción de producción.**
  - El módulo de asignación, permite al jefe de producción asignar tareas teniendo en cuenta la planificación realizada.
  - El módulo de recepción, permite registrar la producción realizada por un empleado de acuerdo a la asignación de trabajo realizado previamente.
- **Reportes de las diferentes etapas de producción.**  
El sistema presenta diferentes reportes de las diferentes etapas de producción, por ejemplo:
  - Estado de un determinado pedido.
  - Asignación de producción.
  - Producción de un empleado.
  - Material recibido, elaborado, devuelto y en posesión de un empleado.

## **5.2. Recomendaciones.**

Luego de haber hecho un estudio de la empresa "Intiwara" y durante el desarrollo sistema, se pudo determinar ciertas necesidades que pueden ser satisfechas realizando las siguientes recomendaciones:

- Desarrollar un sistema integrado que involucre a todos los departamentos de la empresa "Intiwara", teniendo como base el sistema de Gestión del Proceso Productivo, ya que este abarca gran parte de la empresa.
- Desarrollar un portal web con las siguientes características:

- Posibilidad para la comercialización de los productos producidos por la empresa.
- Seguimiento por parte del cliente del avance de su pedido.

## BIBLIOGRAFÍA

- [Pressman, 2010]** Pressman, R. (2010). *Ingeniería del Software un Enfoque Practico*. México: McGraw–Hill.
- [Koontz, 1991]** Koontz, H. y Weihrich, H. (1991) *Elementos de Administración*. México: McGraw–Hill.
- [Fowler, 1997]** Fowler, M. y Kendall, S. (1997). *UML Gota a Gota*. México: Addison Wesley Longman.
- [Wayne, 2005]** Wayne, W (2005) *Investigación de Operaciones Aplicaciones y Algoritmos*. México: Thomson.
- [Seco, 2003]** Seco, A. y Cuadrado, J. (2003) *Análisis y Diseño Estructurado y Orientado a Objetos de Sistemas de Información* México: McGraw–Hill.
- [Jacobson, 2000]** Jacobson, I. Booch, G. y Rumbaugh (2000) *El Proceso Unificado de Desarrollo de Software* Madrid: Pearson Educación.
- [Gil, 2001]** Gil F., Tejedor J., Yague A., Alonso S. y Gutiérrez A. (2001). *Creación de Sitios Web con PHP4*. Madrid: McGraw–Hill.
- [Bennett, 2007]** Bennett S., McRobb S. y Farmer R. (2007). *Análisis y Diseño Orientado a Objetos de Sistemas Usando UML*. Madrid: McGraw–Hill.

### Referencias Web

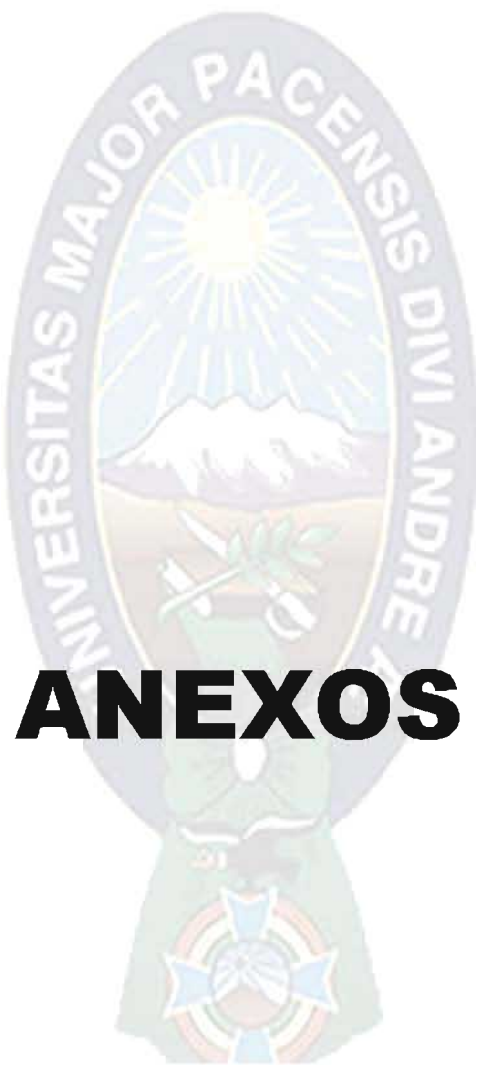
- [wiki Mysql]** <http://es.wikipedia.org/wiki/Mysql>



[capitulo5] [http://catarina.udlap.mx/documentos/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/documentos/marquez_a_bm/capitulo5.pdf)

[monog BD] <http://www.monografias.com/trabajos34/base de datos.shtml>

[wiki segur] [http://es.wikipedia.org/wiki/seguridad\\_inform%C3%A1tica](http://es.wikipedia.org/wiki/seguridad_inform%C3%A1tica)

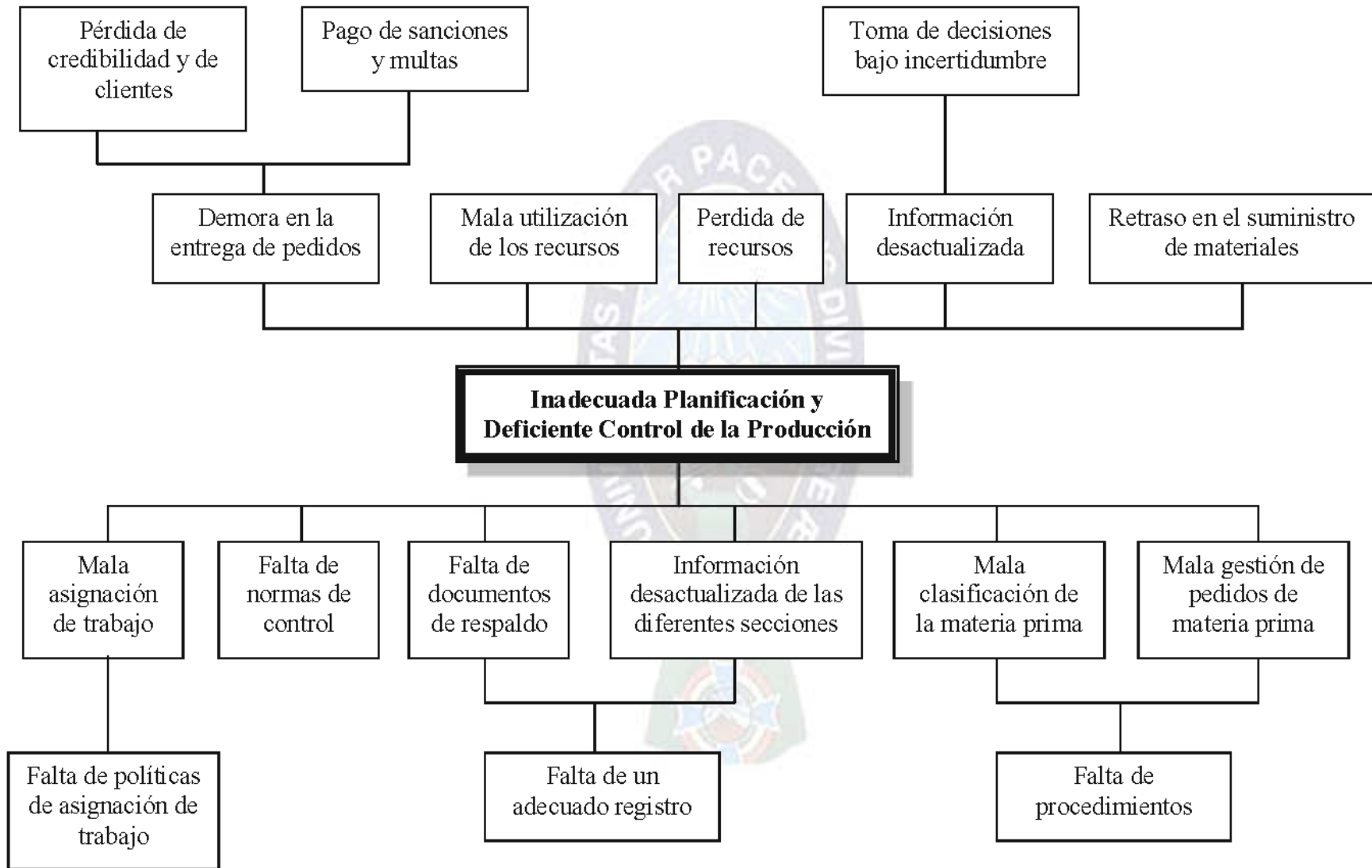


# ANEXOS

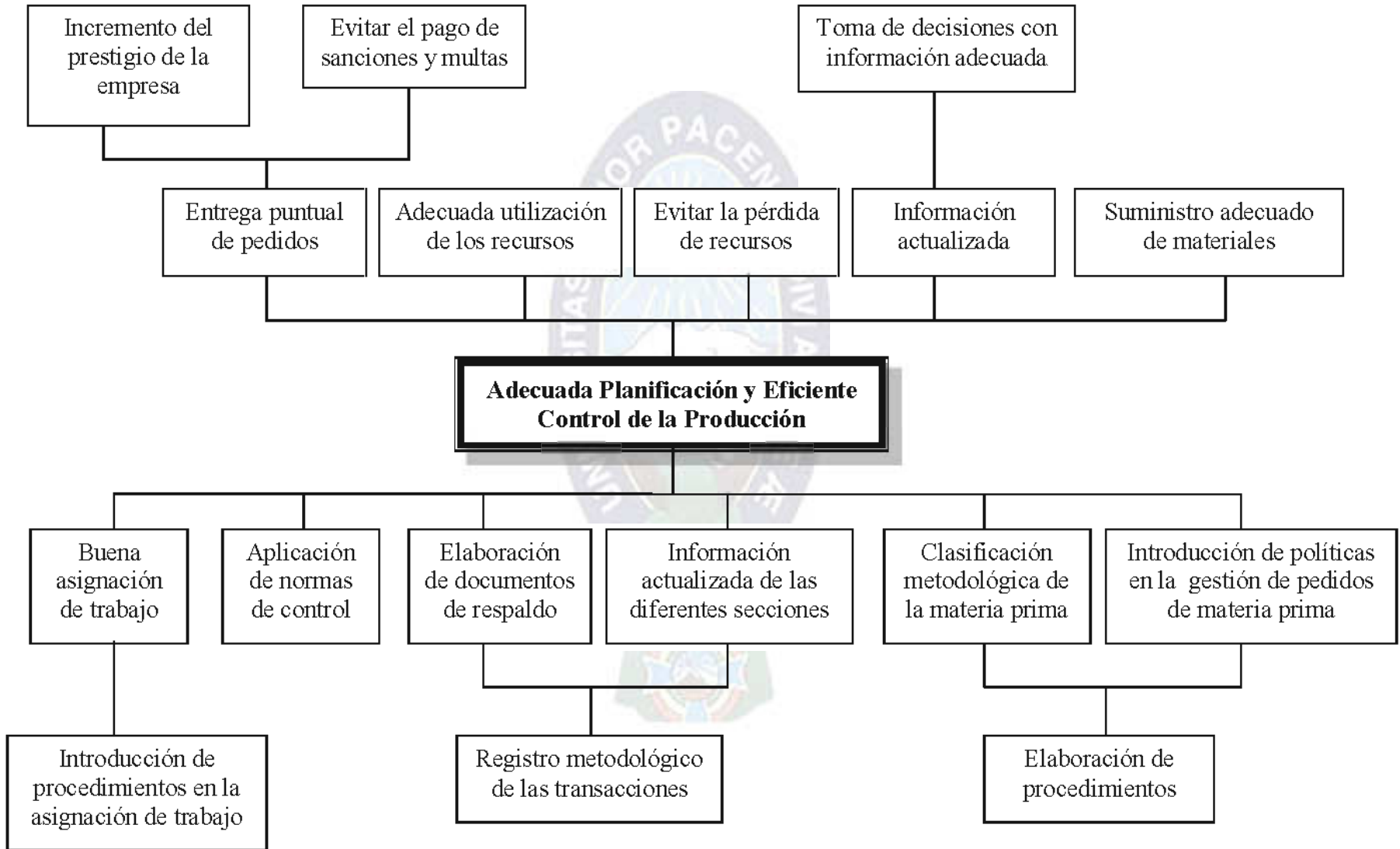
## MATRIZ DEL MARCO LOGICO

Resumen Narrativo de Objetivos	Indicadores Verificables Objetivamente	Medios de Verificación	Supuestos
<p><b>Fin</b> Generar un producto de software sostenible en el tiempo y que sea una herramienta que ayude a consecución de los objetivos</p>	<ul style="list-style-type: none"> <li>- Incremento de la cartera de clientes.</li> <li>- Incremento del prestigio de la empresa.</li> <li>- Consolidación de la fábrica a nivel internacional.</li> </ul>	<ul style="list-style-type: none"> <li>- Libros contables, registro de inventarios.</li> <li>- Aumento de las ventas y de la cartera de clientes a nivel internacional.</li> </ul>	<ul style="list-style-type: none"> <li>- El mantenimiento y actualización del Sistema de Información se realizará periódicamente por el personal adecuado.</li> </ul>
<p><b>Propósito</b> Tener un mejor control de las actividades desarrolladas en el proceso productivo para un uso adecuado de los recursos, y que coadyuve en la toma de decisiones.</p>	<ul style="list-style-type: none"> <li>- Reducción de la pérdida de materiales en un 70%.</li> <li>- Menor pérdida de tiempo y suministro adecuado de materiales</li> <li>- Cumplimiento de los contratos en los tiempos establecidos.</li> </ul>	<ul style="list-style-type: none"> <li>- Aval del tutor y revisor asignados al presente proyecto.</li> <li>- Sistema de Información en plena explotación dentro la empresa.</li> </ul>	<ul style="list-style-type: none"> <li>- Todas las personas involucradas en el desarrollo, explotación y mantenimiento del proyecto colaboran con la consecución de los fines y objetivos.</li> </ul>
<p><b>Componentes</b> Sistema de Información, debidamente instalado, que cuente con los módulos para la</p>	<ul style="list-style-type: none"> <li>- Desarrollo del Sistema de Información en un 100% hasta la segunda quincena del mes de noviembre.</li> </ul>	<ul style="list-style-type: none"> <li>- Revisión del cronograma de actividades.</li> <li>- Manual Técnico, Manual de Operaciones.</li> </ul>	<ul style="list-style-type: none"> <li>- Todas las actividades se desarrollan de manera eficiente.</li> <li>- Los recursos necesarios</li> </ul>
<p><b>Actividades</b> Estudio Preliminar, Análisis, Diseño, Desarrollo, Pruebas, Implementación y Evaluación del Sistema.</p>	<p>Contratación de:</p> <ul style="list-style-type: none"> <li>• 1 Analista de sistemas 800\$ 120 Hrs./hombre.</li> <li>• 2 Programadores 700\$. 200 Hrs./hombre</li> <li>• 1 Auxiliar 300\$. 120 Hrs./hombre</li> </ul> <p><b>Total 1500\$</b></p>	<ul style="list-style-type: none"> <li>- Documentos de respaldo de las actividades realizadas (entrevistas, diagramas, etc.) y los pagos realizados.</li> </ul>	<ul style="list-style-type: none"> <li>- Toda la información requerida será proporcionada por el personal involucrado en los diferentes procesos.</li> <li>- Los recursos necesarios serán proporcionados por la gerencia de la empresa.</li> </ul>

## ARBOL DE PROBLEMAS



## ARBOL DE OBJETIVOS



## DESCRIPCIÓN DE TABLAS

<b>Empleados</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
codigo	varchar	12	Código único asignado a un empleado
nombre	varchar	35	Nombre(s) del empleado
paterno	varchar	25	Apellido paterno del empleado
materno	varchar	25	Apellido materno del empleado
ci	int	10	Numero de cédula de identidad del empleado
fec_nac	date	10	Fecha de nacimiento del empleado
sexo	char	1	Sexo del empleado (M= masculino, F=
direc	varchar	50	Dirección del empleado
Fec ing	date	10	Fecha de ingreso del empleado
seccion	varchar	15	Sección donde trabaja el empleado
sueldo	decimal	8,2	Sueldo básico que percibe el empleado
estado	char	1	Estado en que se encuentra el empleado
tel_fijo	varchar	8	Número telefónico fijo del empleado
tel_cel	varchar	8	Número de teléfono celular del empleado

<b>Usuarios</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
codigo	varchar	12	Código único asignado a un usuario
contraseña	varchar	20	Contraseña de acceso del usuario
Fec alta	date	10	Fecha de alta de un usuario
Fec modi	date	10	Fecha de modificación de la contraseña
nivel	int	4	Numero de nivel de acceso
estado	char	1	Estado de un usuario
Nro vis	int	20	Número de visitas realizadas por un usuario

<b>Productos</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro pro	int	10	Número de producto
Nom pro	varchar	25	Nombre del producto
Tip_pro	varchar	20	Tipo de producto
Maq_pro	varchar	12	Maquina que produce el producto
Observ	varchar	50	Observaciones acerca del producto
Pro cos	decima	8,2	Precio pagado por la costura del producto
Pro bor	decima	8,2	Precio pagado por el bordado del producto
peso	decima	8,2	Peso aproximado del producto
filas	int	4	Número de filas aproximadas de un producto
tiempo	int	4	Tiempo aproximado de producción de un

<b>asi tej (Asignación de Tejidos)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro_asi	int	10	Número de asignación
cliente	varchar	15	Nombre del cliente
cod emp	varchar	12	Código de empleado
Fec asi	date	10	Fecha de asignación
Est asi	char	3	Estado en que se encuentra la asignación

<b>rec tej (Recepción de Tejidos)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro rec	int	10	Número de recepción
cod_emp	varchar	12	Código de empleado
Fec rec	date	10	Fecha de recepción del tejido
Nro_pag	int	10	Número de pago realizado por el tejido
Peso_rec	decima	8,2	Peso total de los productos tejidos

<b>Det_asi (Detalle de Asignación)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro deas	int	10	Numero de detalle de asignación
Nro asi	int	10	Número de asignación
Nro ped	int	10	Número de pedido
Nro depe	int	10	Número de detalle de pedido
Mod asi	varchar	20	Modelo asignado
Col asi	varchar	15	Color del modelo asignado
Tip asi	varchar	15	Tipo del modelo asignado
sas	int	4	Cantidad de la talla S asignado
mas	int	4	Cantidad de la talla M asignado
las	int	4	Cantidad de la talla L asignado
xlas	int	4	Cantidad de la talla XL asignado
otas	int	4	Cantidad de otras tallas asignadas
est	char	3	Estado en que se encuentra la asignación
observ	varchar	70	Observaciones relativas a la asignación

<b>Det_rec (Detalle de Recepción)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro dere	int	10	Numero de detalle de recepción
Nro rec	int	10	Número de recepción
Nro deas	int	10	Número de detalle de asignación
Mod rec	varchar	20	Modelo producido
Col rec	varchar	15	Color del modelo producido
Tip rec	varchar	15	Tipo del modelo producido
sre	int	4	Cantidad de la talla S producido
mre	int	4	Cantidad de la talla M producido
lre	int	4	Cantidad de la talla L producido
xlre	int	4	Cantidad de la talla XL producido
otre	int	4	Cantidad de otras tallas producidas

<b>Pedidos</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro ped	int	10	Número de pedido
cliente	varchar	20	Nombre del cliente
Fec rec	date	10	Fecha de recepción del pedido
Fec ent	date	10	Fecha de entrega del pedido
estado	char	3	Estado en que se encuentra el pedido
material	char	3	Estado del material requerido por el pedido

<b>Det_pedido (Detalle de Pedido)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro_det	int	10	Numero de detalle de pedido
Nro ped	int	10	Número de pedido
Mod_ped	varchar	20	Modelo pedido
Col ped	varchar	15	Color del modelo pedido
Tip ped	varchar	15	Tipo del modelo pedido
spe	int	4	Cantidad de la talla S pedido
mpe	int	4	Cantidad de la talla M pedido
lpe	int	4	Cantidad de la talla L pedido
xlpe	int	4	Cantidad de la talla XL pedido
otpe	int	4	Cantidad de otras tallas pedido
est	char	3	Estado en que se encuentra el pedido

<b>Inventario</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
num	int	5	Numero de ítem introducido
codigo	varchar	10	Código de la materia prima
peso	decima	6,2	Peso de la materia prima
conos	int	4	Número de conos de materia prima
fecha	date	10	Fecha de registro



<b>Ing mat (Ingreso de Materia Prima)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro ing	int	4	Número de Ingreso
Nro fac	varchar	20	Número de factura de compra de Materia Prima
Fec ing	date	10	Fecha de Ingreso de Materia Prima
peso	decima	8,2	Peso total de la Materia Prima

<b>Det ing (Detalle de Ingreso de Materia Prima)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro_ing	int	6	Número de Ingreso
Cod_ing	varchar	15	Código de Materia Prima
Peso_ing	decima	8,2	Peso de Materia Prima
conos	int	4	Número de conos ingresados

<b>Ent_mp (Entrega de Materia Prima)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro ent	int	6	Numero de entrega de materia prima
Fec ent	date	10	Fecha de entrega de materia prima
Cod emp	varchar	12	Código del empleado que recibe la materia
Nro con	int	4	Número de conos de materia prima
peso	decima	8,2	Peso entregado de materia prima
Nro ord	int	10	Número de asignación

<b>dev mp (Devolución de Materia Prima)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
Nro_dev	int	6	Numero de devolución de materia prima
Fec_dev	date	10	Fecha de devolución de materia prima
Cod_emp	varchar	12	Código del empleado que devuelve la materia
Nro_con	int	4	Número de conos devueltos con materia prima
peso	decima	8,2	Peso devuelto de materia prima

<b>Medidas</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
nro	int	10	Numero de medida
modelo	varchar	25	Nombre del modelo
cliente	varchar	25	Nombre del cliente
tipo	varchar	20	Tipo de modelo
observ	varchar	50	Observaciones de las medidas

<b>Det_med (Detalle de las Medidas)</b>			
<b>Campo</b>	<b>Tipo</b>	<b>Lon</b>	<b>Descripción</b>
nro	int	6	Número de medida
item	varchar	20	Item relacionado a las tallas
xs	int	4	Medida de la talla XS
s	int	4	Medida de la talla S
m	int	4	Medida de la talla M
l	int	4	Medida de la talla L
xl	int	4	Medida de la talla XL
xxl	int	4	Medida de la talla XXL
xxxl	int	4	Medida de la talla XXXL